



Эволюция архитектур нейронных сетей от перцептрана к GPT

Руденко Марина Анатольевна

Кандидат технических наук, доцент кафедры компьютерной инженерии и моделирования
Директор Центр искусственного интеллекта и анализа больших данных ФГАОУ ВО
«Крымский федеральный университет имени Вернадского» Симферополе, Россия.



План лекции:

1. Трансформация парадигм ИИ
2. Рождение идеи - Перцептрон и его ограничения
3. Многослойные сети и обратное распространение
4. Специализированные архитектуры - CNN и RNN
5. Трансформеры - революция в архитектуре
6. Эра больших языковых моделей - от BERT к GPT
7. Будущие тренды ИИ

«Самое захватывающее в эволюции — это то, что она никогда не останавливается. Каждый финиш — это новый старт»

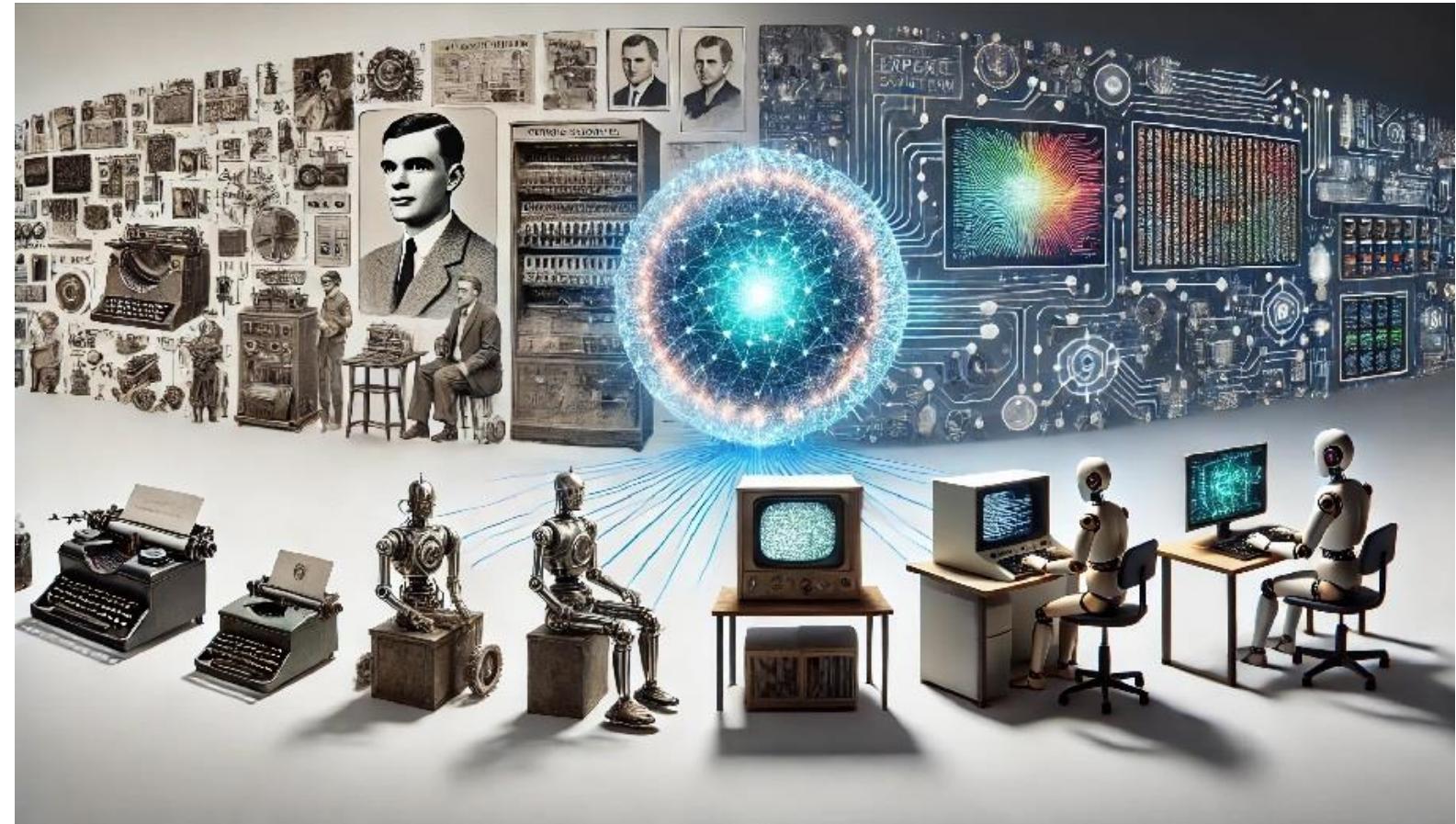
Айзек Азимов





1. Трансформация парадигм ИИ

*Ничего в современных
нейросетях не имеет смысла
без понимания их
архитектурной эволюции*

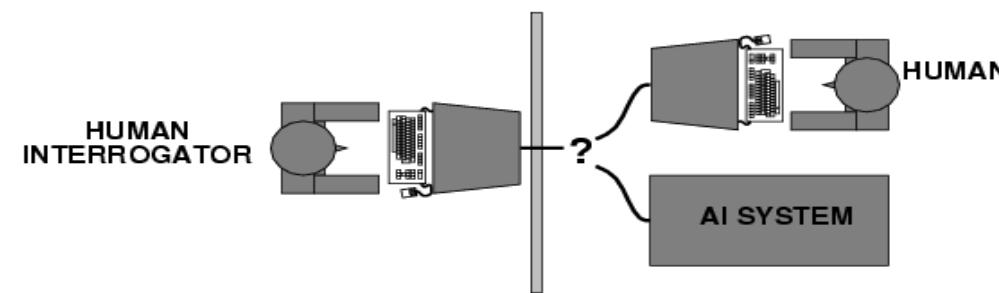




1950-ые: СОЗДАНИЕ ОБЛАСТИ ИИ



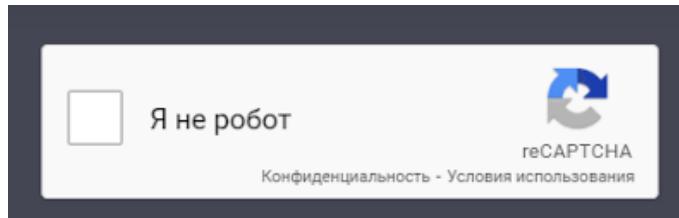
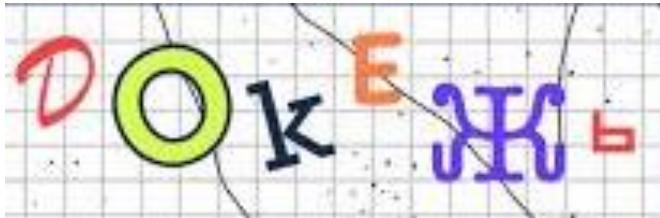
А. Тьюринг (1950) "Computing machinery and intelligence":
«Может ли машина мыслить?» → «Может ли машина вести себя
интеллектуально?»
Оперативный тест для проверки интеллектуального поведения:



Алан Тьюринг, 1950



Тесты Тьюринга сегодня:

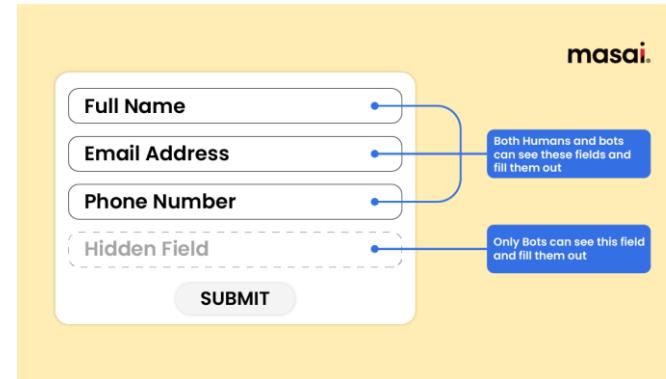
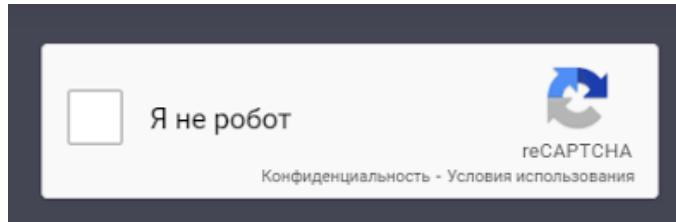


Тесты Тьюринга сегодня:

1. Тест Тьюринга с открытыми возможностями. Это расширенный тест, применимый к роботам(Стивен Харнард в 1991 году).
2. Обратный тест Тьюринга. Это еще один вариант теста, частным случаем которого оказывается капча (от английского **CAPTCHA**, completely automated public Turing test to tell computers and humans apart)
3. Конкурс на премию Лебнера, который проводился с 1991 по 2020 год. Разработчикам первой программы, прошедшей тест Тьюринга, организаторы обещали вручить \$25 тыс



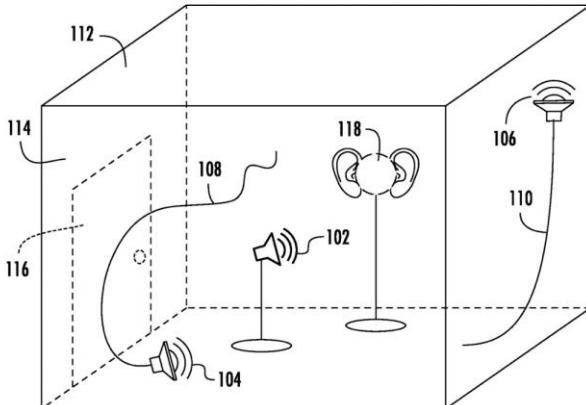
Тесты Тьюринга будущее:



Капча точно не исчезнет, но бороться ей придется с сильно поумневшими ИИ. Вот только справится ли она?

Системы, которые успешно анализируют активность на сайте и выявляют опасности на основе машинного обучения — [GuardDuty](#) от Amazon и [Watson](#) for Cybersecurity.

Аудио-CAPTCHA



основанные на Ньютоновской физике

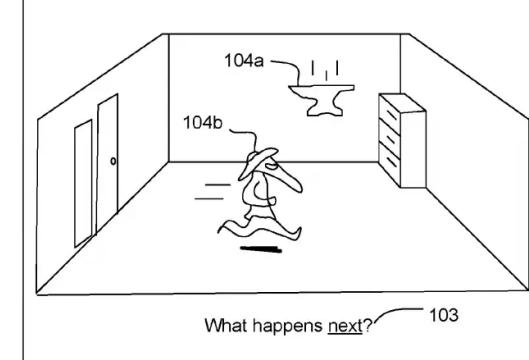


FIG. 1A

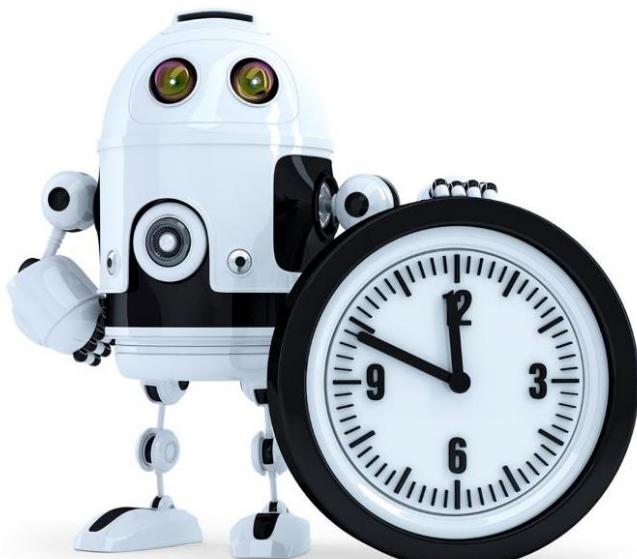


FIG. 1B





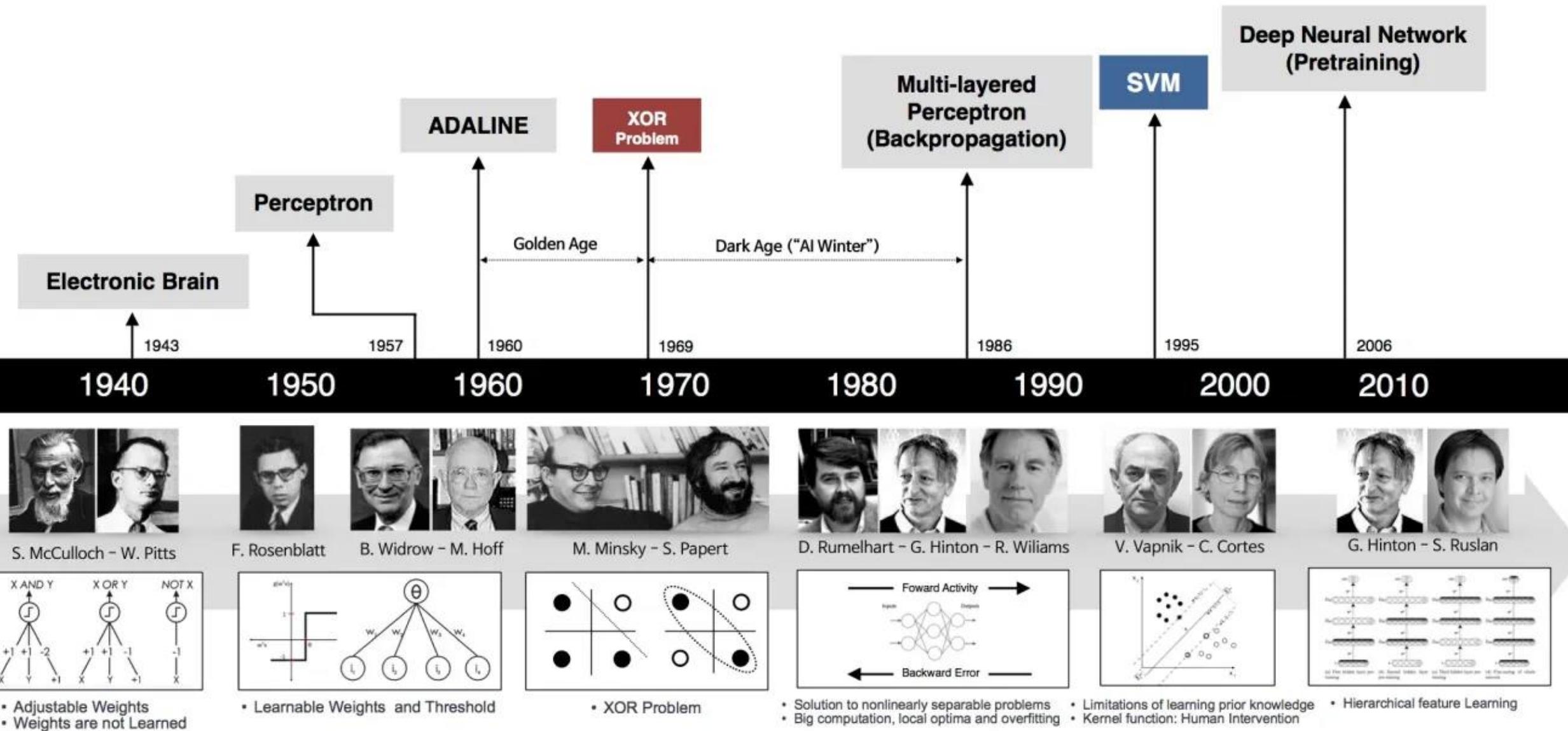
Краткая история ИИ



1943	Первая работа по ИИ Мак-Каллока и Питтса: модель, состоящая из искусственных нейронов
1950	«Computing Machinery and Intelligence» А.Тьюринга
1956	Дартмутский семинар Маккарти: придумано название "Artificial Intelligence" для новой научной области
1952- 69	Look, Ma, no hands! Ранний энтузиазм, большие ожидания
1950е	Ранние программы ИИ: Шахматная программа Самюэла; Logic Theorist Ньюэлла и Саймона, Geometry Engine Гелентера доказательства теорем
1965	Алгоритм Робинсона для логического рассуждения, обладающий свойством полноты
1966-73	Столкновение ИИ с реальностью: вычислительная сложность.
1969—79	Начало разработок систем, основанных на знаниях (СОЗ). Экспертные системы DENDRAL и MYSIN.
1980--	ИИ становится индустрией
1986--	Возвращается популярность нейронных сетей
1987--	ИИ превращается в науку
1995--	Появление интеллектуальных агентов



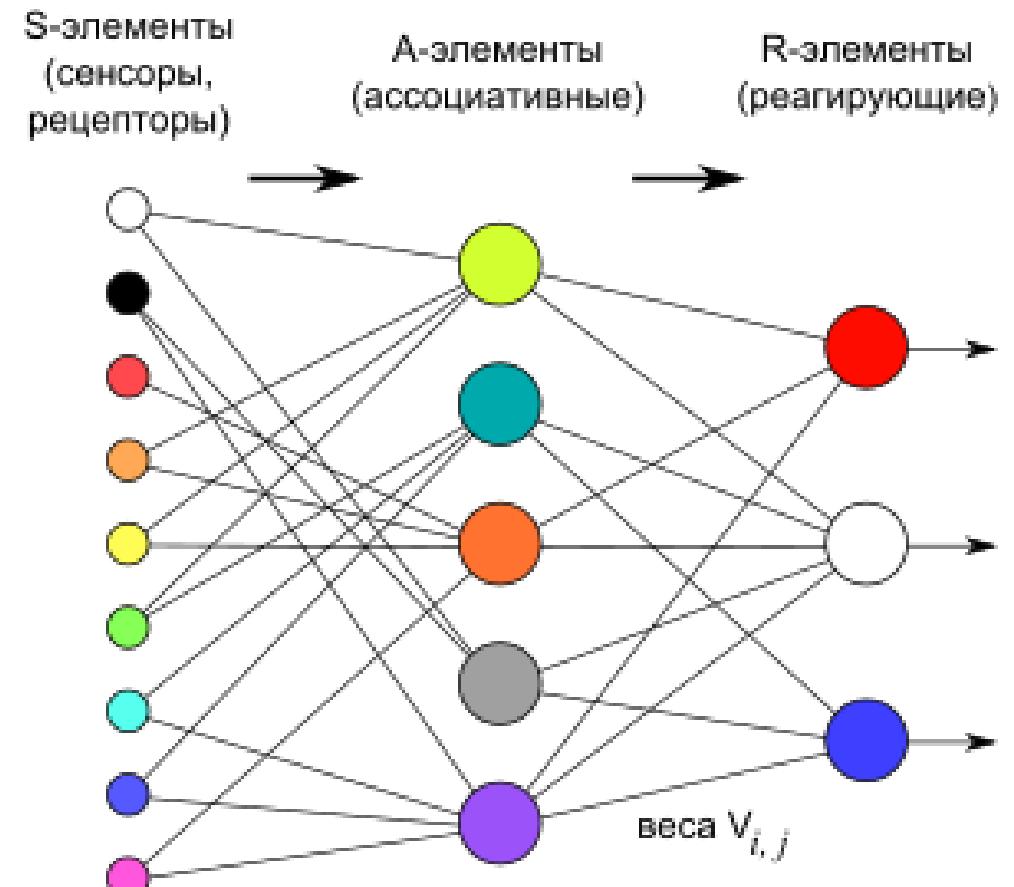
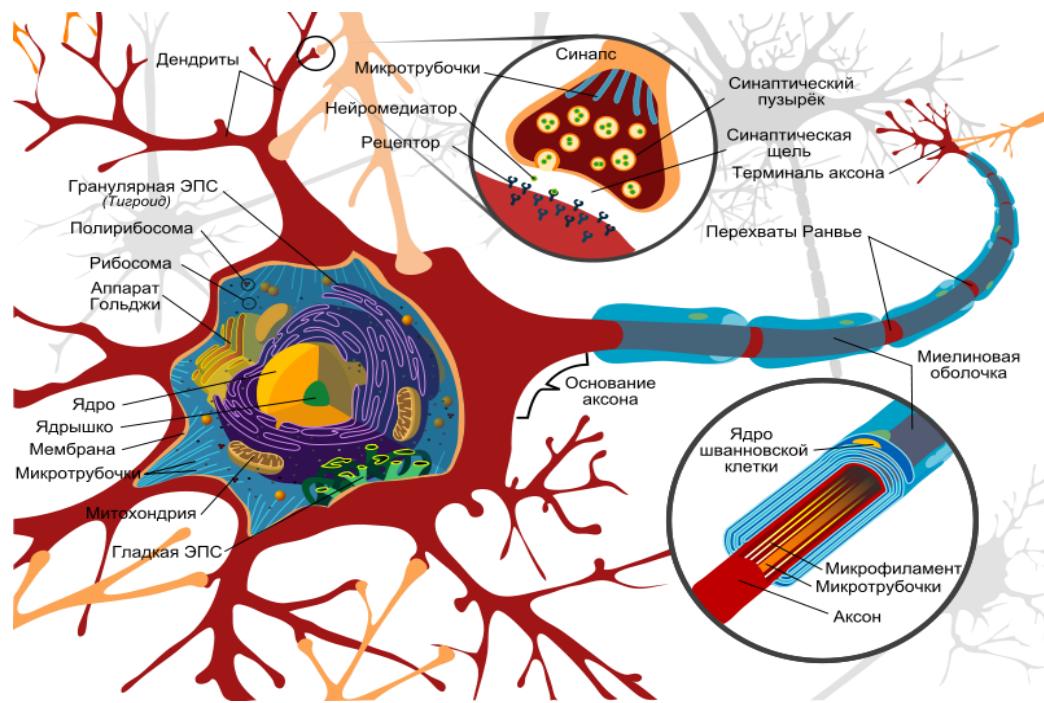
Краткая история нейронных сетей





2. Рождение идеи - Перцептрон и его ограничения

Искусственная нейронная сеть — математическая модель, построенная по принципу организации и функционирования биологических нейронных сетей.

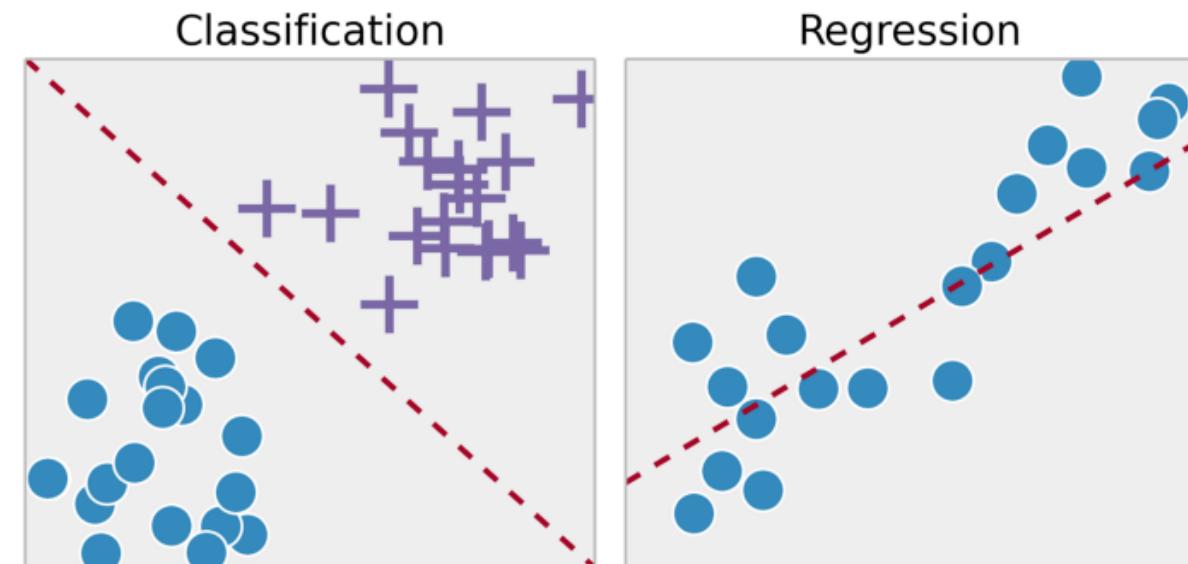
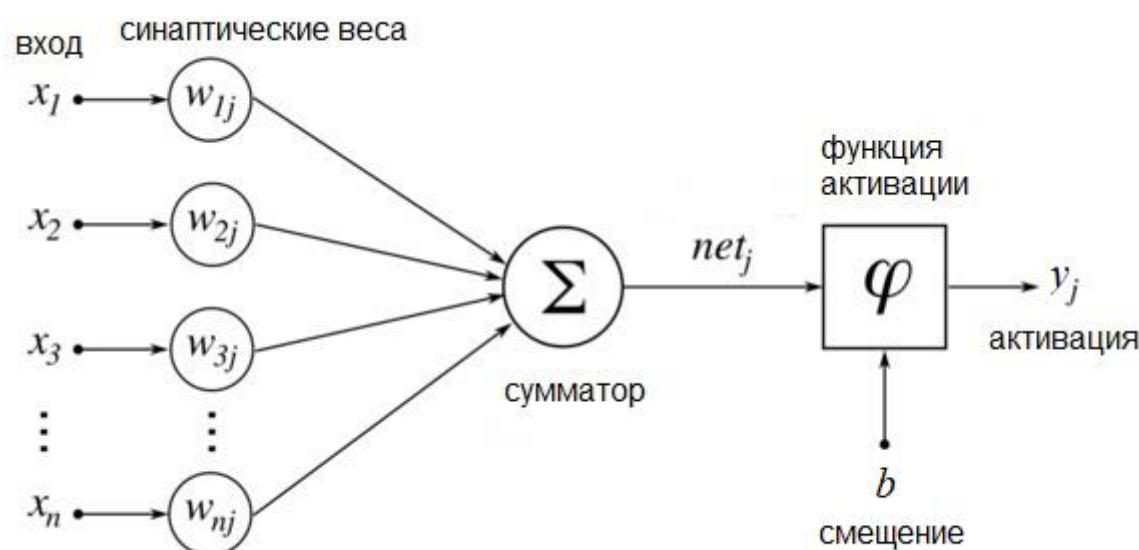




2. Рождение идеи - Перцептрон и его ограничения

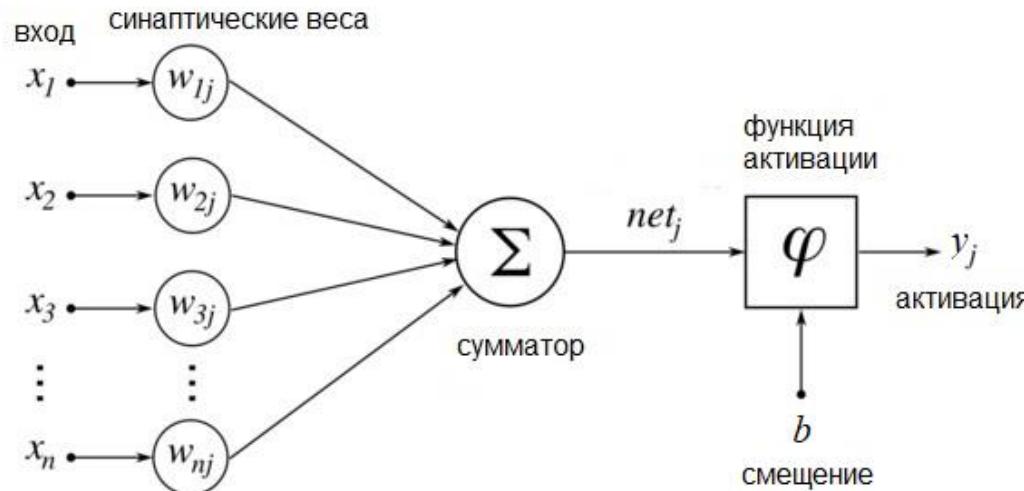
Я бы смог решить вашу задачку... если бы она была прямой линией

Нейронные сети принадлежат к классу алгоритмов, обучающихся с учителем (supervised learning), и решает типовые задачи этого класса:





Нервная клетка vs Искусственный нейрон



ИН – формализованная модель биологического нейрона, предложенная в 1943 году У. Маккалоком и У.Питтсон.

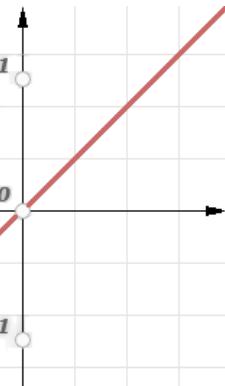
Он работает следующим образом:

- Нейрон получает на вход вектор x
- Его компоненты умножаются на соответствующий вес и складываются. Также прибавляется смещение b .
- К взвешенной сумме применяется функция активации.

$$Y = \varphi(\sum w_i x_i + b)$$



Функции активации



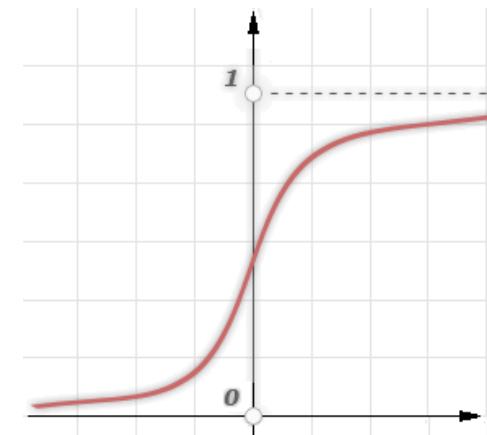
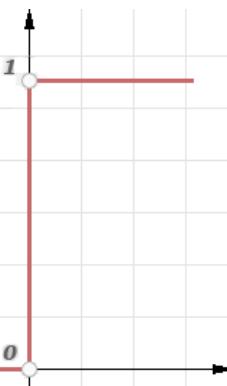
Линейная

- Выходы сети являются линейными комбинациями входов

Пороговая

Эта функция использовалась в оригинальной модели ИН.

- + имеет центрированный аналог ($\text{sign } x$)
- не дифференцируема



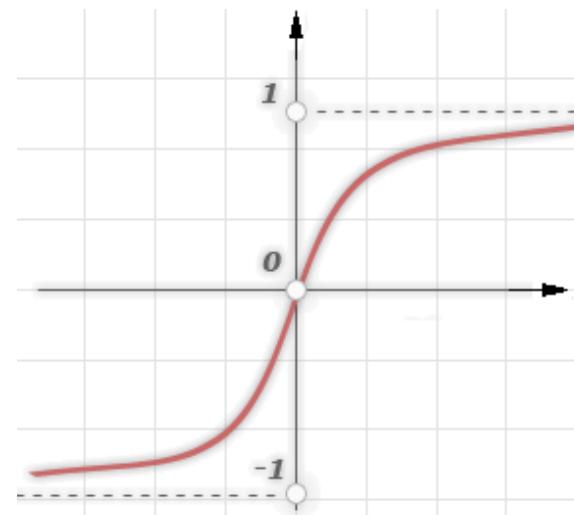
Сигмоида

$$f(x) = \frac{1}{1 + e^{-\alpha x}}$$

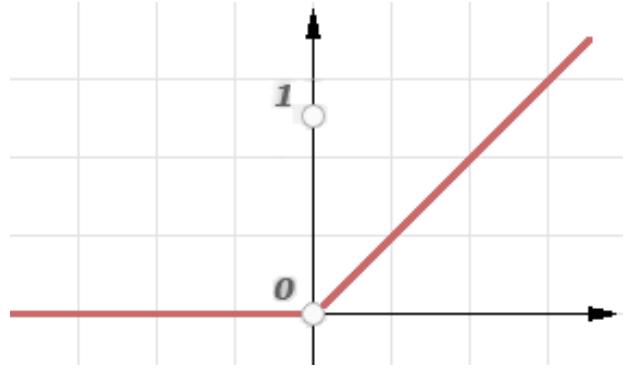
Долгое время считалась функцией, лучше всего описывающей работу нейрона.

- + дифференцируема
- имеет порог насыщения

$$\text{Tanh}(x) = \tanh\left(\frac{\alpha x}{2}\right) = \frac{1 - e^{-\alpha x}}{1 + e^{-\alpha x}}$$



- + дифференцируема
- + центрована
- имеет порог насыщения



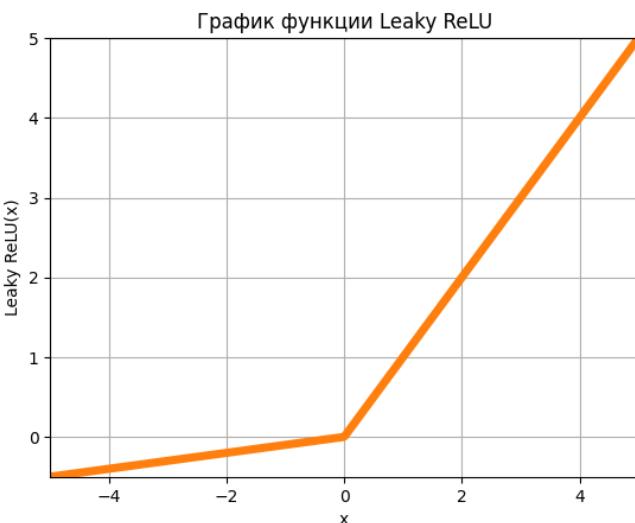
- ReLU (rectified linear unit)

$$f(x) = \max(0, x)$$

- В настоящее время самая широко используемая функция активации в силу своей простоты.
- Также недавние исследования показывают, что она правильнее описывает работу биологических нейронов
- + дифференцируема
- + не имеет порога насыщения
- + быстро вычисляется
- не центрована
- чувствительна к инициализации

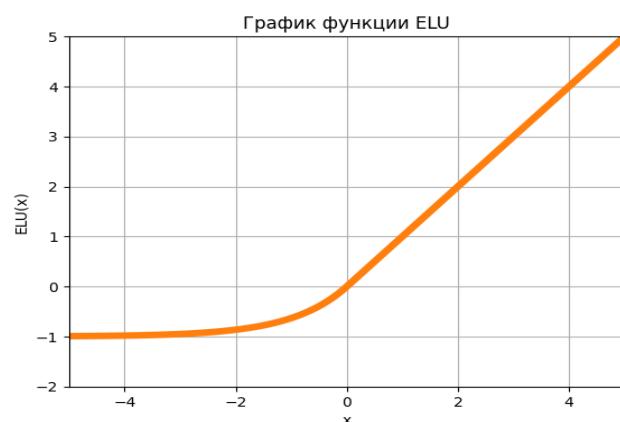
Leaky ReLU (Rectified Linear Unit)

Преимуществом Leaky ReLU является устойчивость к "умиранию" нейронов и лучшая сходимость в процессе обучения, что приводит к более быстрому и точному обучению нейронных сетей.

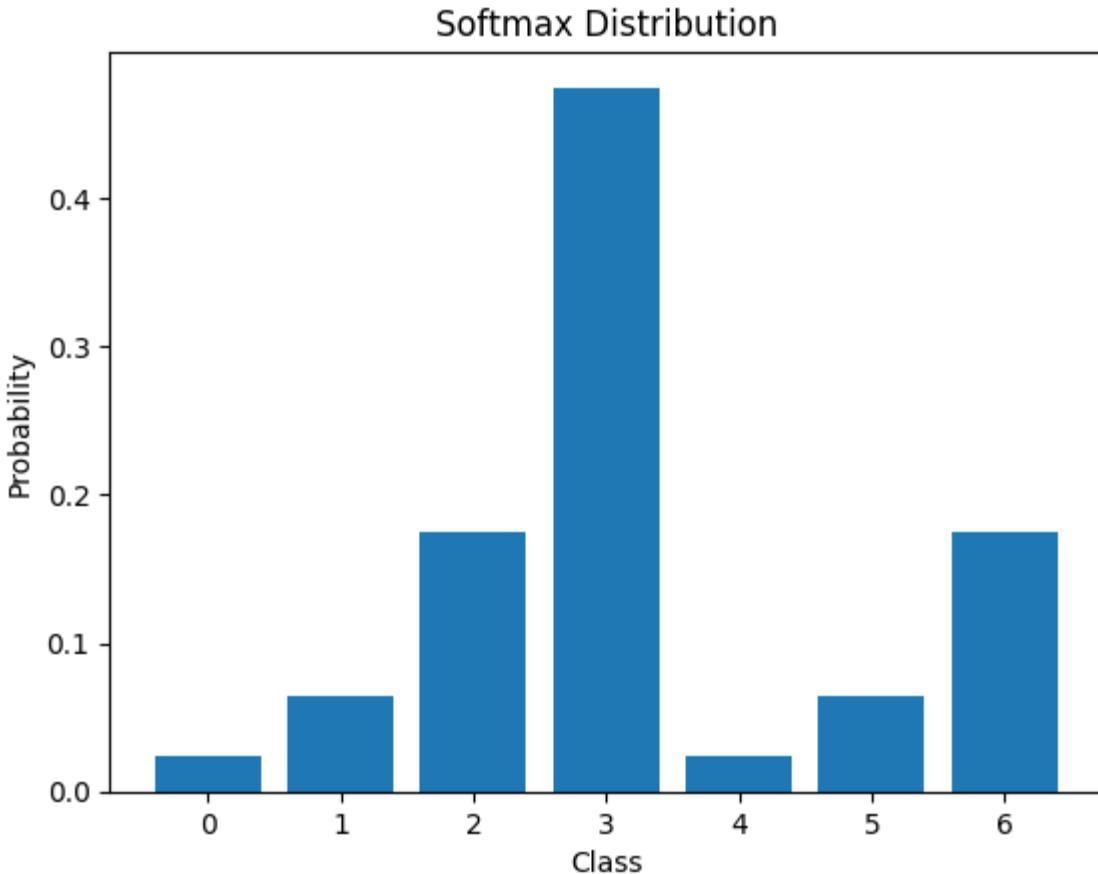


$$\text{Leaky Relu}(x) = \max(\alpha * x, x)$$

ELU (Exponential Linear Unit) - функция активации, которая представляет собой измененную версию ReLU (Rectified Linear Unit), которая помогает ускорить обучение глубоких нейронных сетей и справляется с проблемой "мертвых нейронов" (dead neurons).



$$\text{ELU}(x) = \begin{cases} x, & \text{if } x > 0 \\ \alpha(e^x - 1), & \text{if } x \leq 0 \end{cases}$$



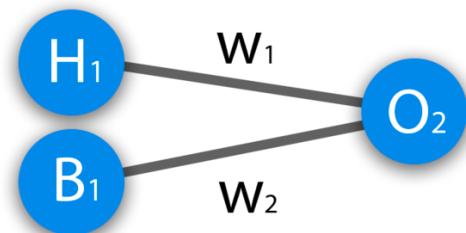
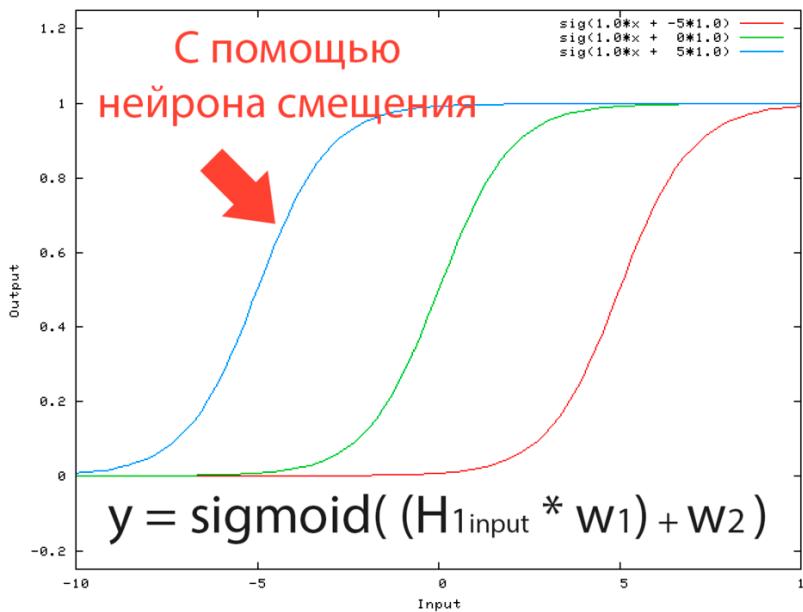
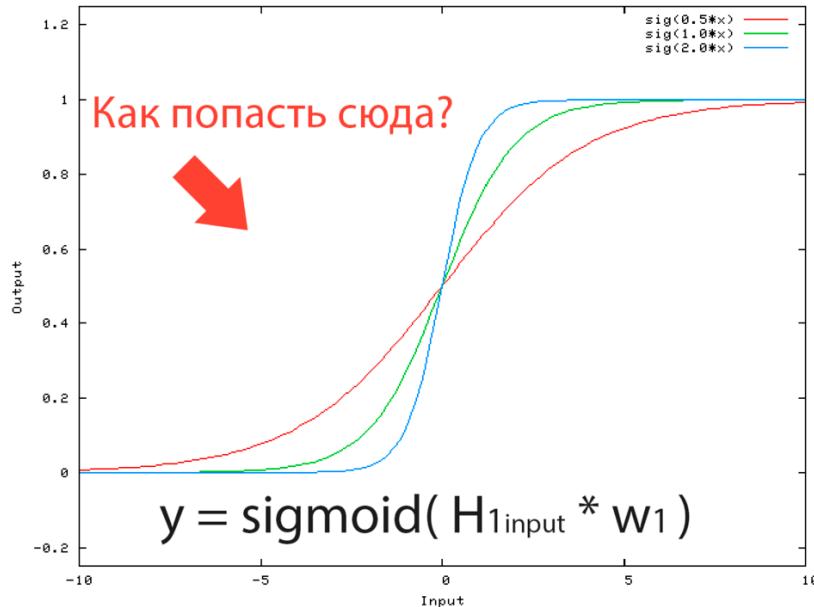
Функция Softmax используется для преобразования вектора значений в вероятностное распределение, которое суммируется до 1. Она особенно полезна в многоклассовой классификации, где необходимо определить вероятности для каждого класса.

Формула функции Softmax выглядит следующим образом:

$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^k e^{z_k}}$$



Зачем нужно смещение



Замечание: сдвиг b можно также считать отдельным нейроном, на который всегда подается значение 1. Такой нейрон называется **нейроном смещения**.



Разделяющая гиперплоскость

Я бы смог решить вашу задачку... если бы она была прямой линией

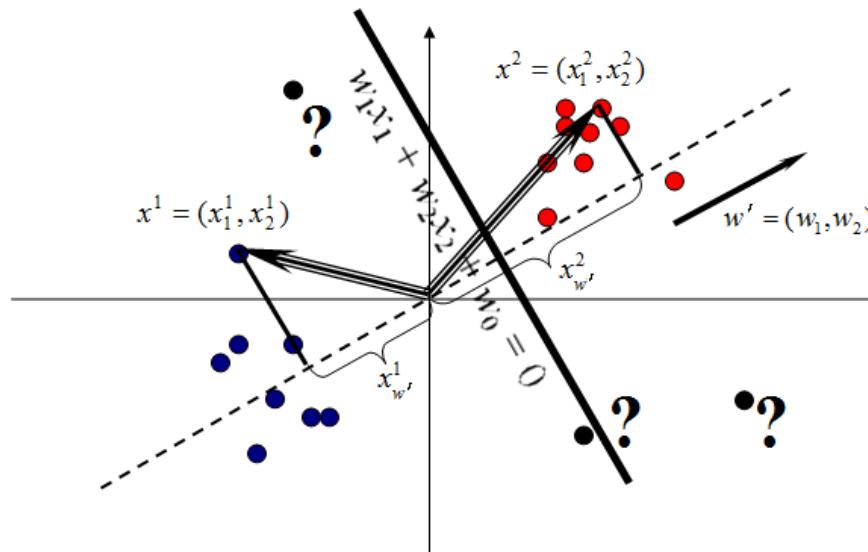
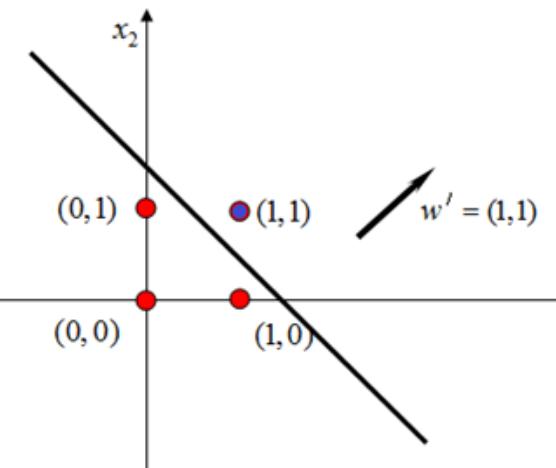
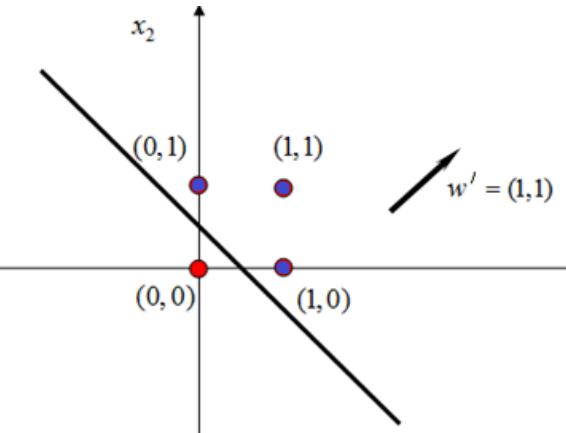
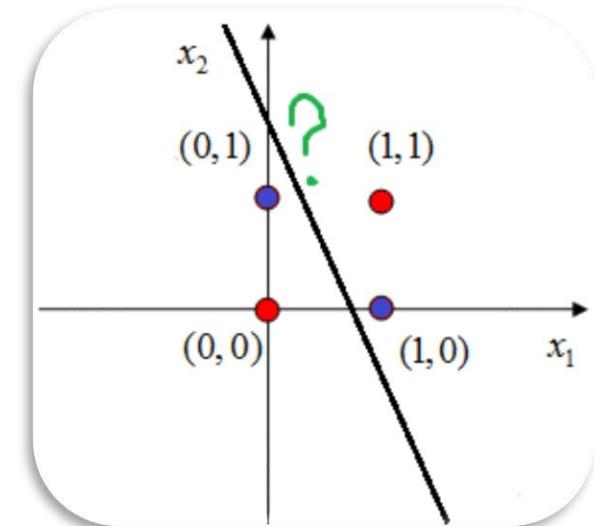


Рис. 3. Геометрическая интерпретация задачи разделения элементов множества на два класса в двумерном пространстве.

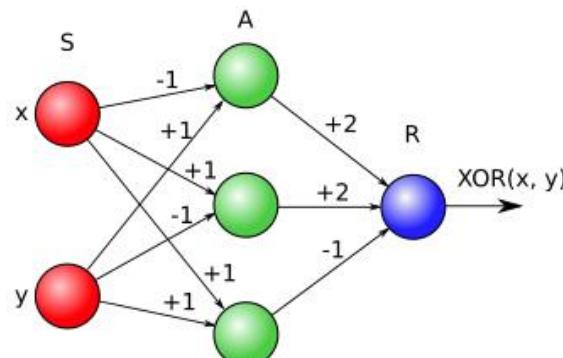
В задаче классификации однослойный персептрон строит в R^n гиперплоскость (или поверхность, если функция активации нелинейна), разделяющую объекты на 2 класса.



Научное сообщество на долгое время потеряла интерес к нейронным сетям после выхода в 1969 году статьи **Марвина Минского и Сеймура Паперта**, в которой утверждалось, что персептрон не способен обучаться функции XOR.



3. Многослойные сети и обратное распространение



Перцептрон - одна из первых моделей нейронных сетей, предложенная в 1957 году Ф. Розенблаттом как средство решения задач классификации.

$$y = f(\sum w_2 f(\sum w_1 x))$$

ПРАВИЛА ОБУЧЕНИЯ ПЕРСЕПТРОНА

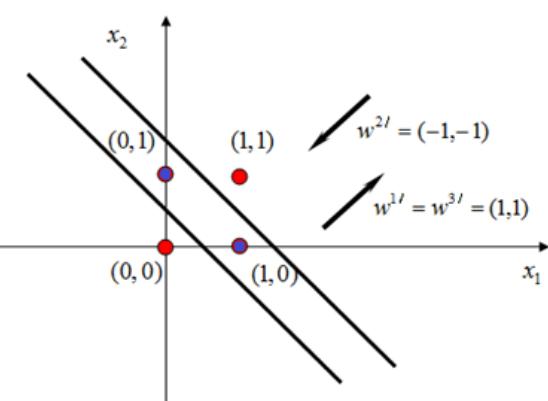
Персептрон – нейронная сеть прямой передачи сигнала с бинарными входами и бинарной пороговой функцией активации.

Правило обучения Розенблatta в общем случае является вариантом правил обучения Хебба, формирующих симметричную матрицу связей, и в тех же обозначениях имеет вид:

$$w_{ij}(t+1) = w_{ij}(t) + \eta \cdot (t_i - y_j) \cdot x_i$$

где η – коэффициент обучения, $0 < \eta < 1$,

t_j – эталонные или целевые значения.



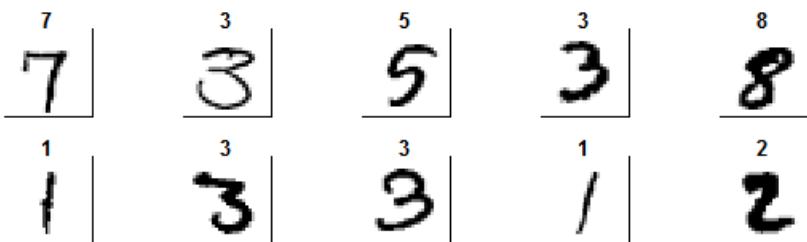


2. Перцептрон и его обучение

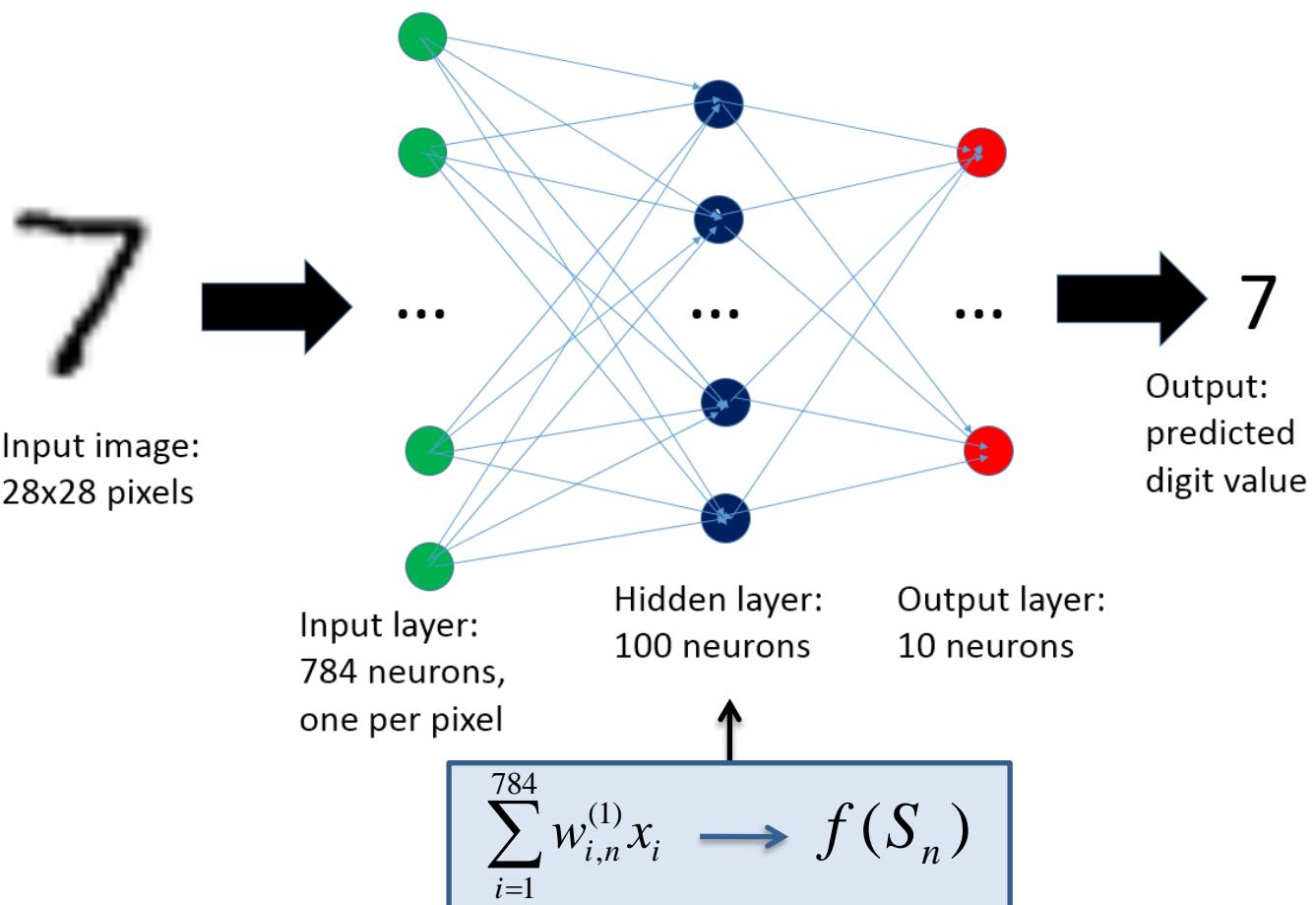
Обучающая выборка

D

x_1	\tilde{y}_1
x_2	\tilde{y}_2
⋮	
x_N	\tilde{y}_N



Выборка – набор размеченных входных векторов (т.е. таких, для которых известен правильный ответ), по которому производится настройка сети.





2. Перцептрон и его обучение

Функция потерь

Функция потерь – функция, по значению которой можно оценить работу сети.

Две наиболее часто используемых функции потерь:

- среднеквадратичная ошибка (MSE):

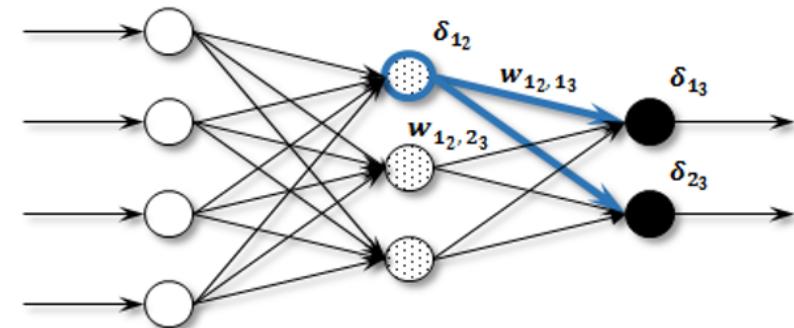
$$E = \frac{1}{2} \sum_{i=1}^N (y_i - \tilde{y}_i)^2$$

- логистическая (log loss):

$$E = -\frac{1}{N} \sum_{i=1}^N (\tilde{y}_i \cdot \log(y_i) + (1 - \tilde{y}_i) \cdot \log(1 - y_i))$$

Наиболее распространенный метод обучения нейронной сети – **метод обратного распространения ошибки**.

Основная идея этого метода состоит в распространении сигналов ошибки от выходов сети к её входам, в направлении, обратном прямому распространению сигналов в обычном режиме работы.

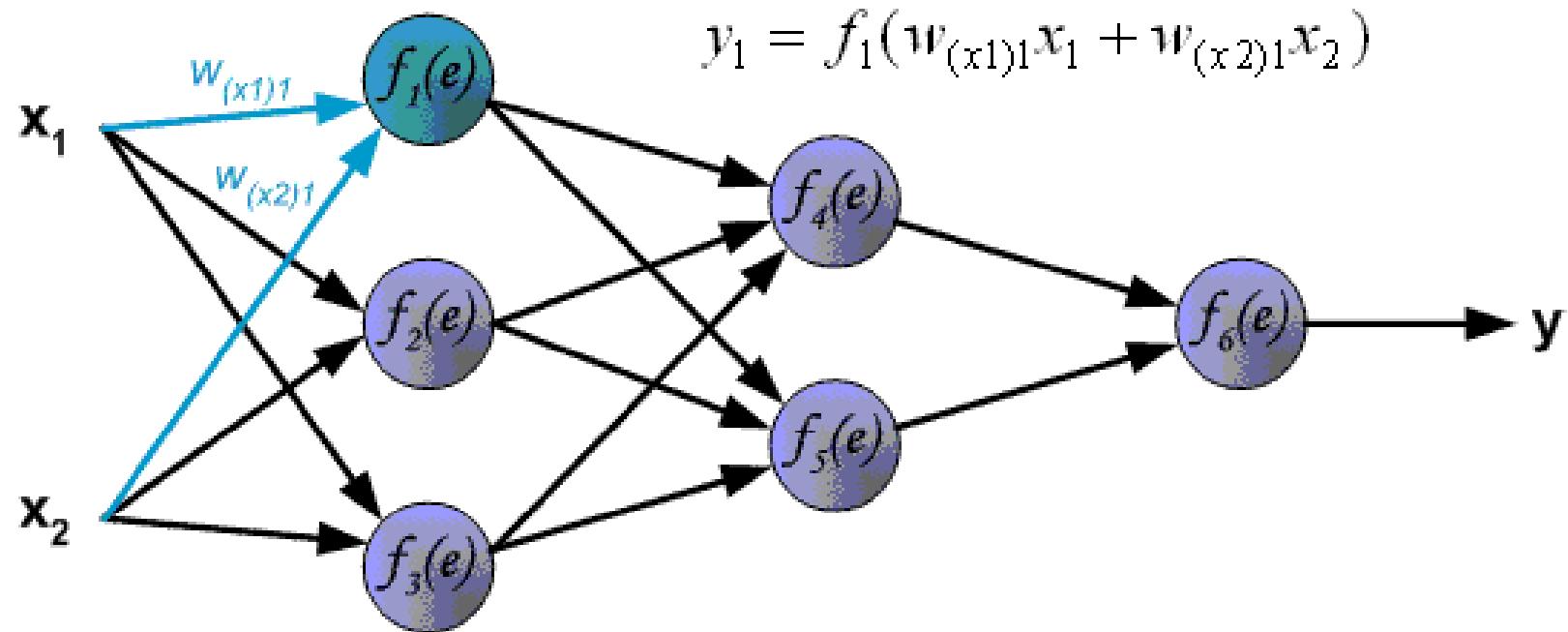


- Для последнего слоя: $\delta_j = (y_j - \tilde{y}_j)f'(S_j)$
- Для внутренних слоев: $\delta_j = \left(\sum_k \delta_k w_{jk} \right) f'(S_j)$
- Для всех: $\Delta w_{ij} = -\eta \delta_j y_i$
 $w_{ij} = w_{ij} + \Delta w_{ij}$



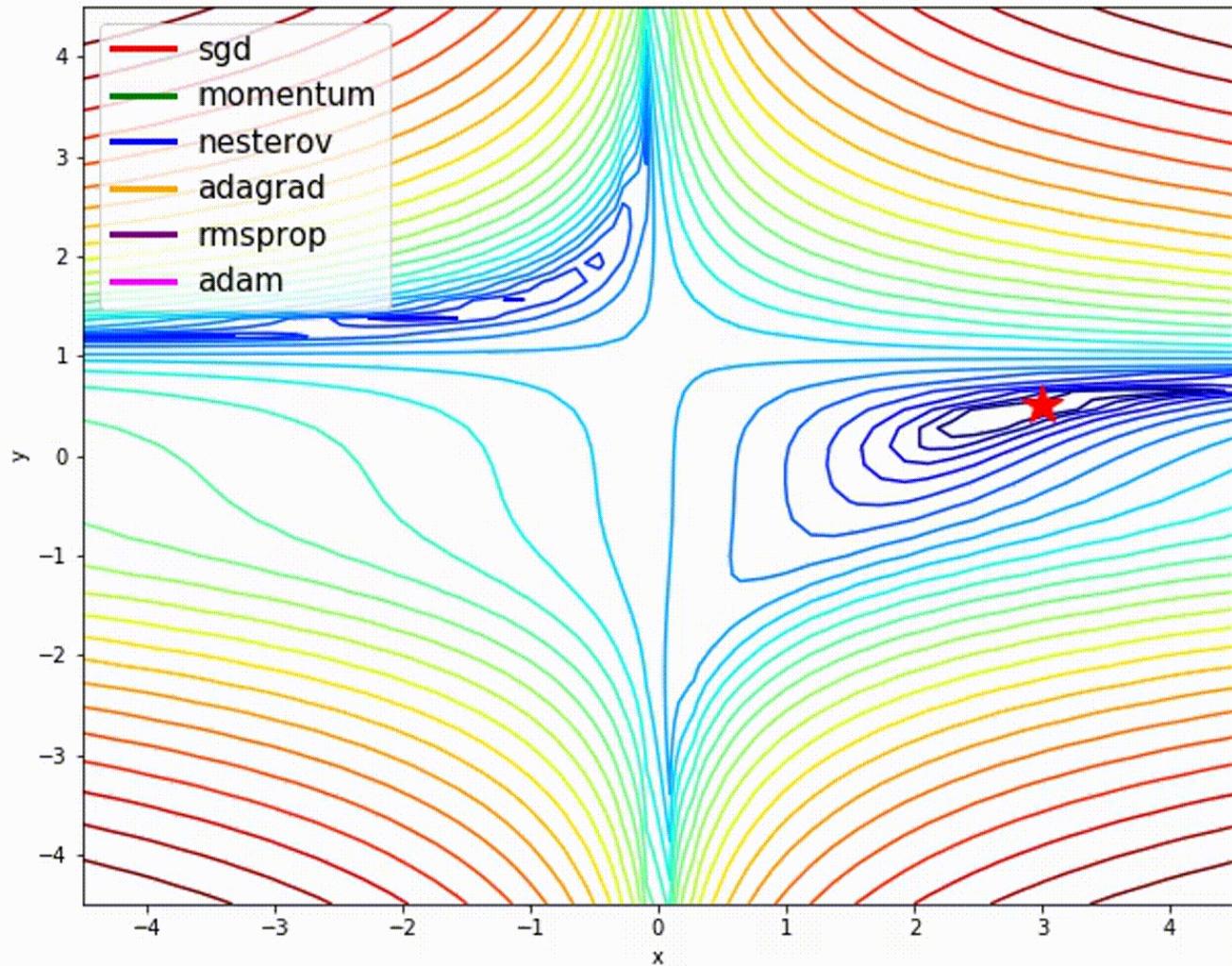
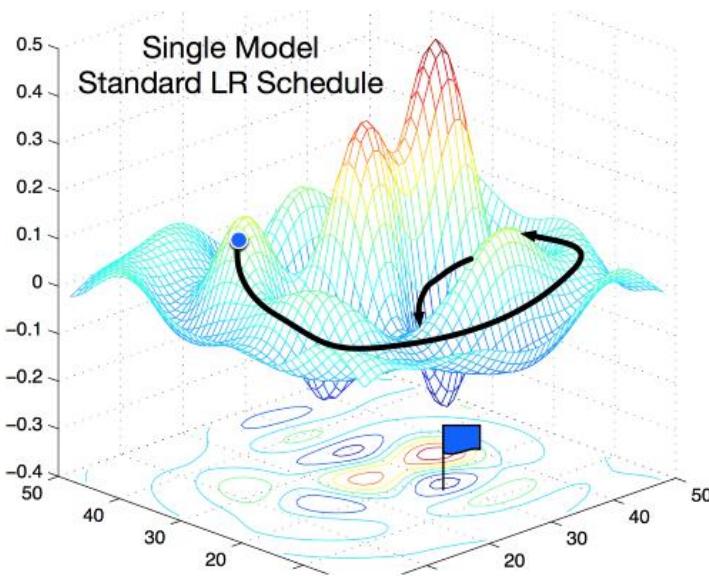
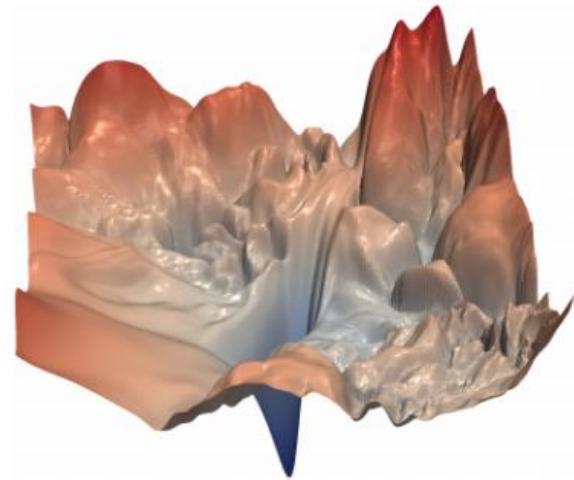
Метод обратного распространения ошибки

FP





Роль оптимизаторов





Проблемы обучения

- Паралич сети – сеть перестает обучаться.
- Переобучение
- Недообучение

Причины:

- затухающий градиент
- взрывающийся градиент
- неправильный выбор гиперпараметров

Контроль обучения

- Кросс-валидация
- Регуляризация
 - штраф за большие веса
 - dropout
 - batch norm
- Работа с обучающей выборкой



Достижения нейронных сетей

От перцептрана к трансформеру: дорога длиною в 70 лет

1970-е: ЭКСПЕРТНЫЕ СИСТЕМЫ



ЕСЛИ
(Температура тела > 37,5°C)
И
(Дискомфорта в глазах нет)
И
...
И
(Головная боль отсутствует)
ТО
ПРОСТУДА

VS

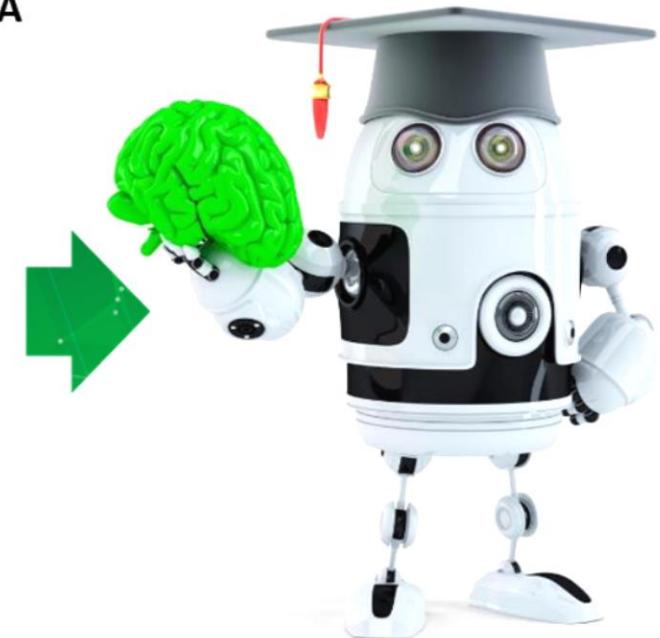
1980-е: МАШИННОЕ ОБУЧЕНИЕ

ДАННЫЕ ДЛЯ АЛГОРИТМА

ОБЪЕКТЫ

- Мед. книжка
- Анализы
- Диагноз

ПРОСТУДА



1997, IBM DEEP BLUE ПРОТИВ КАСПАРОВА



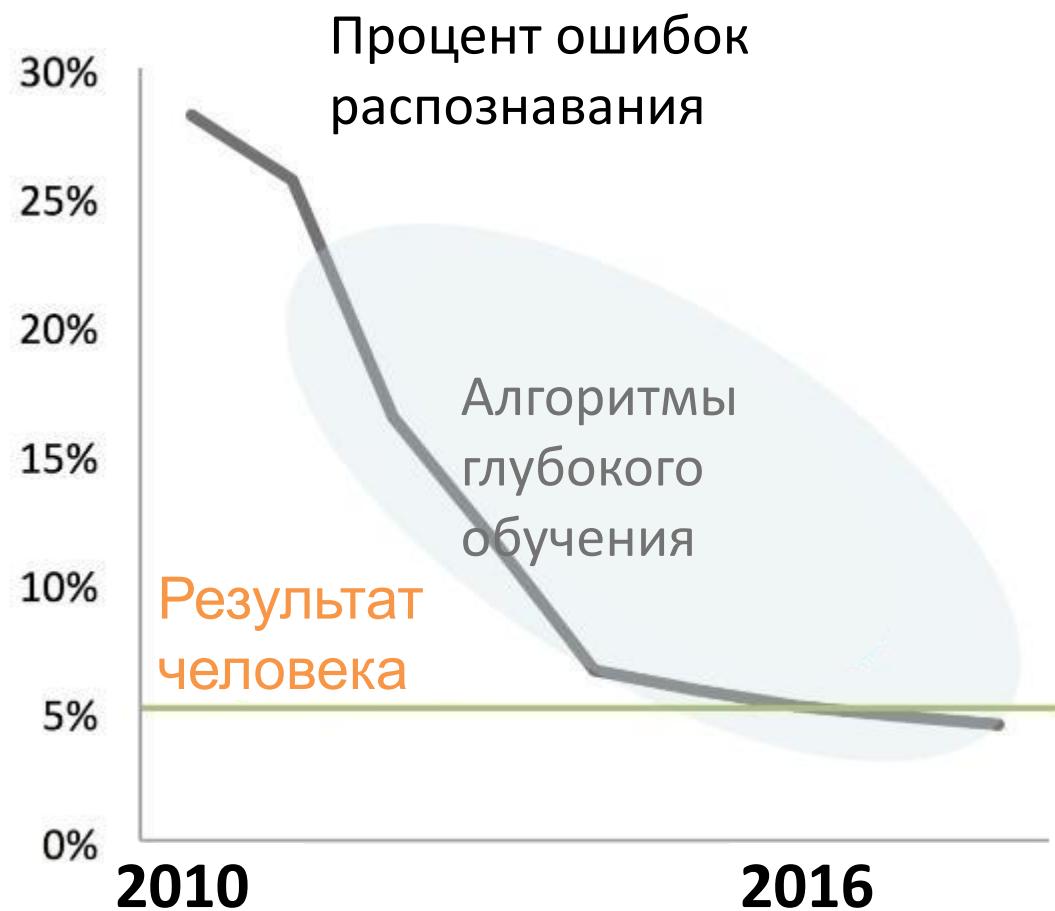
2004, GRAND DARPA CHALLENGE 2009, GOOGLE CAR



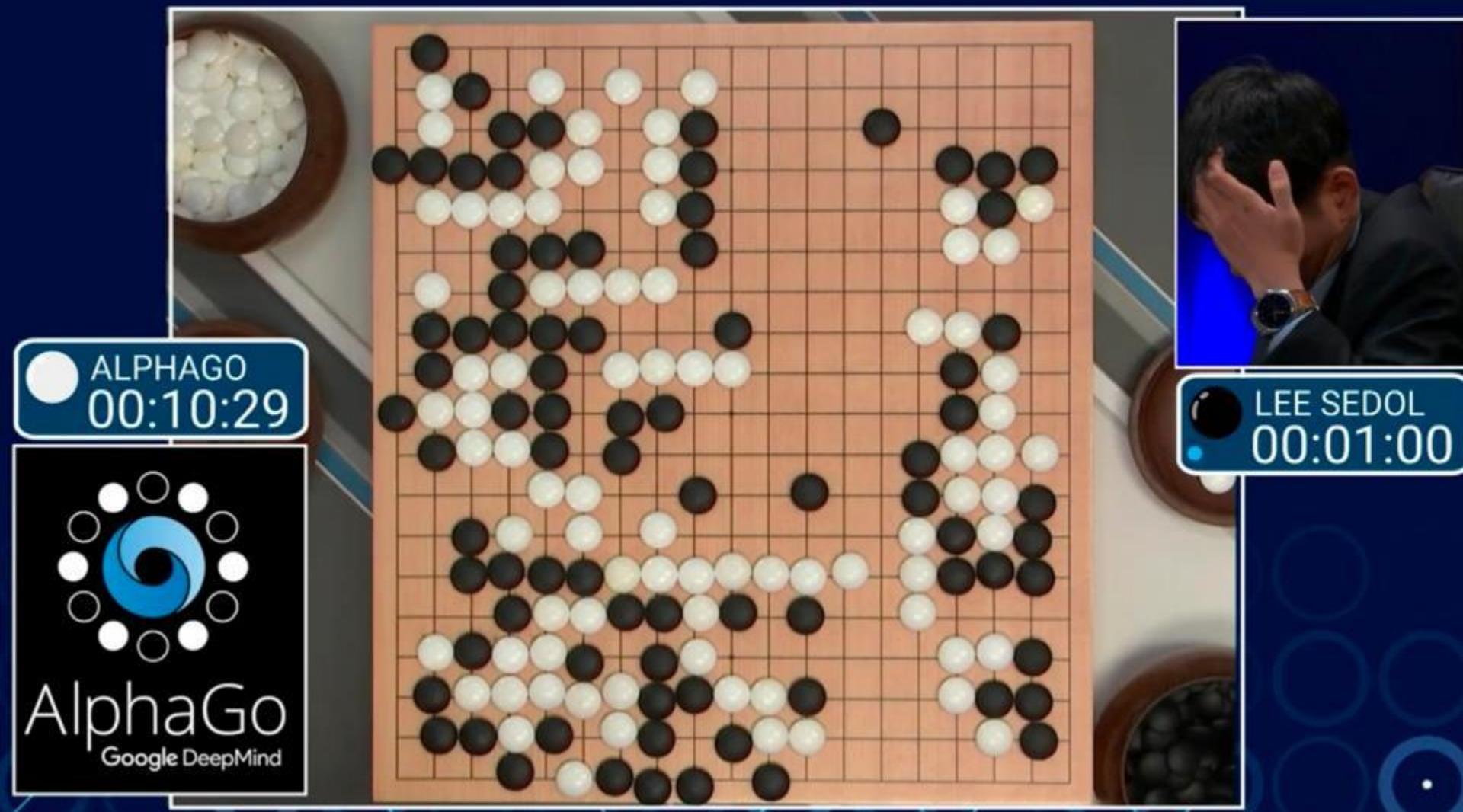
2011, IBM WATSON И «СВОЯ ИГРА»



2011 – 2015, ГЛУБОКОЕ ОБУЧЕНИЕ



2016, ALPHAGO GOOGLE DEEPMIND ПОБЕДИЛ В ГО



2017, OPENAI И ПОБЕДА В DOTA 2



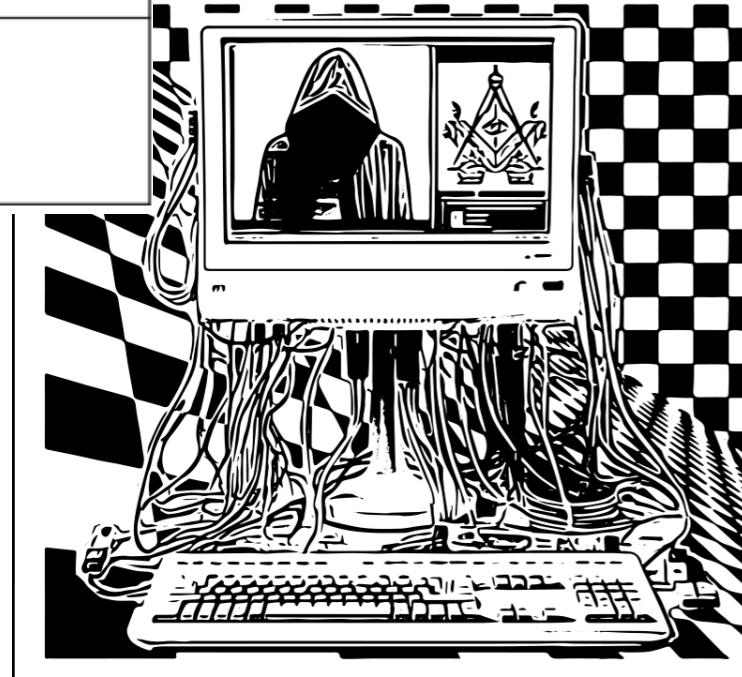
Достижения машинного обучения:





Что такое Искусственный интеллект?

Все разнообразие взглядов на ИИ можно разделить на 4 категории :

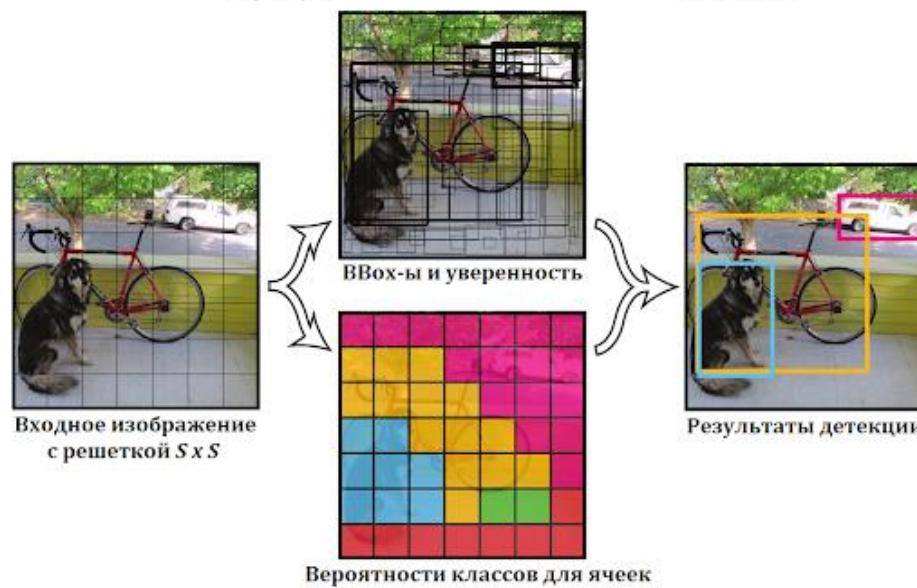
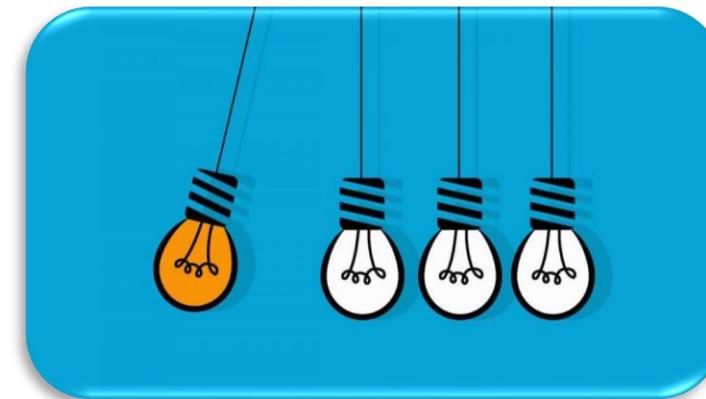
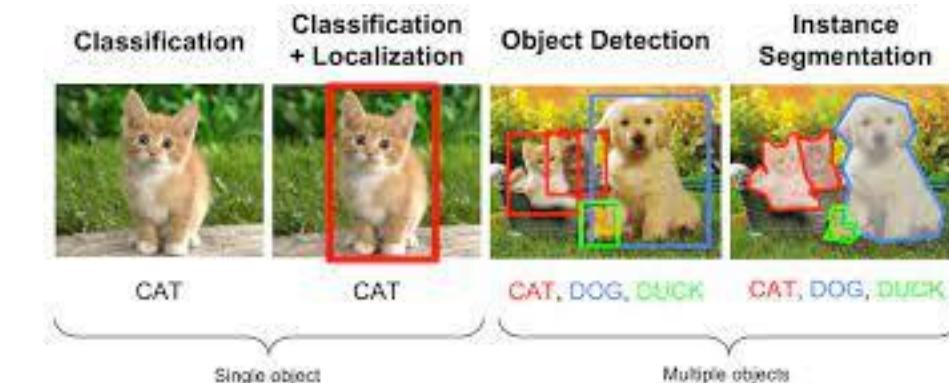
Системы, которые думают подобно людям	Системы, которые думают рационально	
Системы, которые действуют подобно людям	Системы, которые действуют рационально	
		

Искусственный интеллект и выход за границы

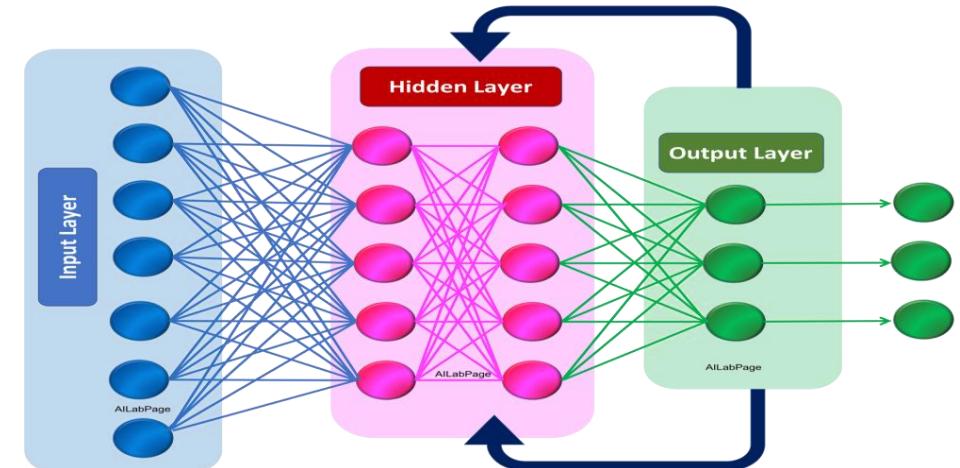
Сейчас системы искусственного интеллекта используют человеческий опыт, знания и даже язык. Но все это ограничивает способность ИИ к эволюции. ИИ переизобретает логику и понимание мира. И создает свое, непонятное нам, сознание. Которое стремительно эволюционирует.



4. Специализированные архитектуры - CNN и RNN

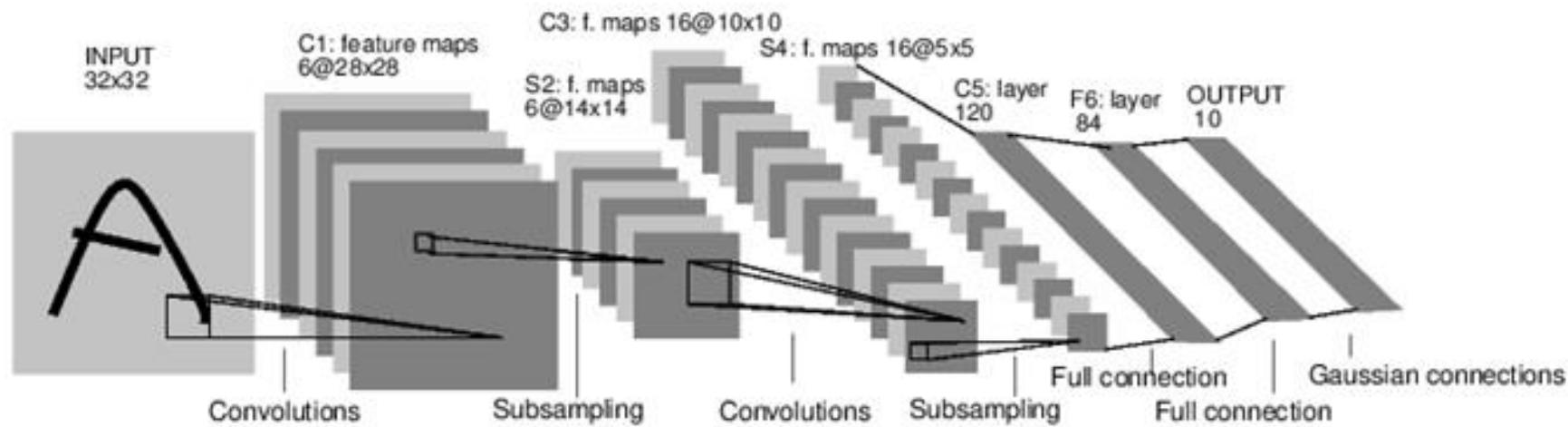


Recurrent Neural Networks





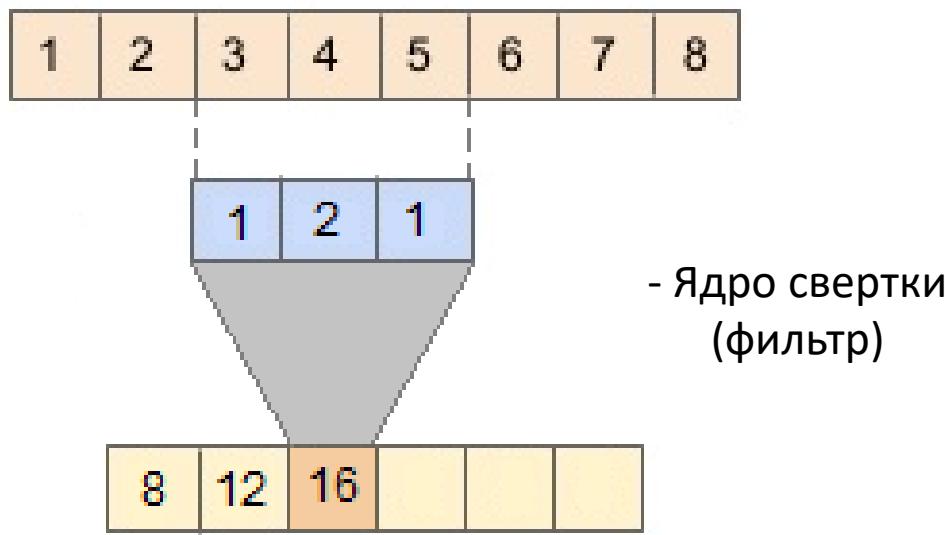
Свёрточные нейронные сети



A Full Convolutional Neural Network (LeNet)

Сверточная нейронная сеть (*CNN*) — специальная архитектура нейронных сетей, предложенная Яном Лекуном в 1988 году и нацеленная на эффективное распознавание изображений. Идея заключается в чередовании сверточных слоев и субдискретизирующих слоев.

Свертка



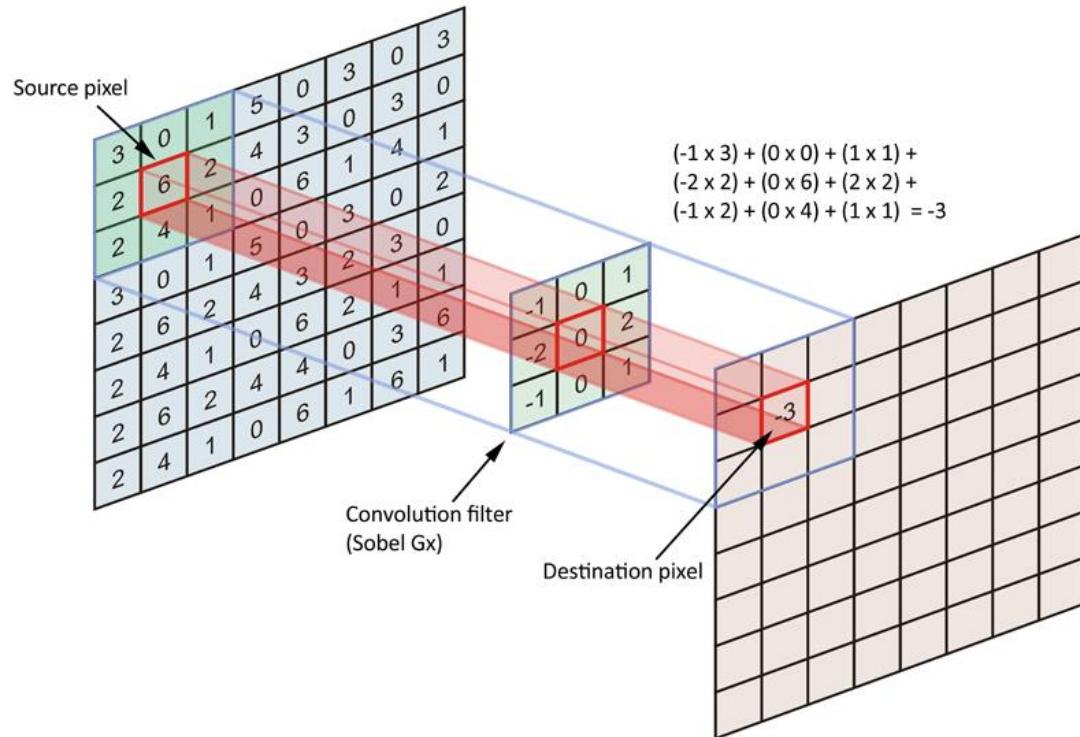
Свертка – операция, применяемая к двум массивам, которая заключается в следующем:

- фильтр «скользит» по входному массиву, и каждый элемент выходного массива равен скалярному произведению фильтра и соответствующей области входного массива.

$$n[k] = \sum_{i=-w}^w m[k+i]a[-i]$$



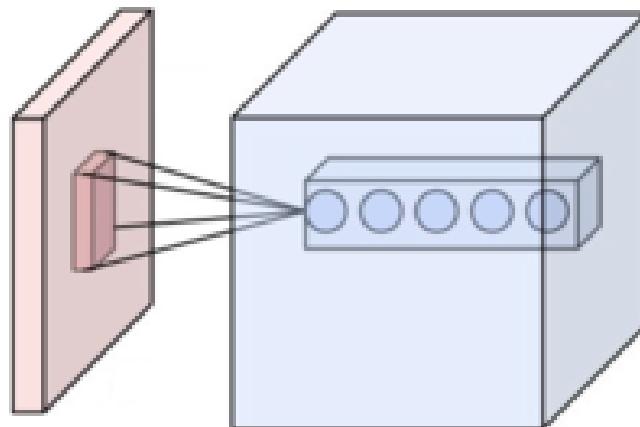
Двумерная свертка



3 ₀	3 ₁	2 ₂	1	0
0 ₂	0 ₂	1 ₀	3	1
3 ₀	1 ₁	2 ₂	2	3
2	0	0	2	2
2	0	0	0	1

12	12	17
10	17	19
9	6	14

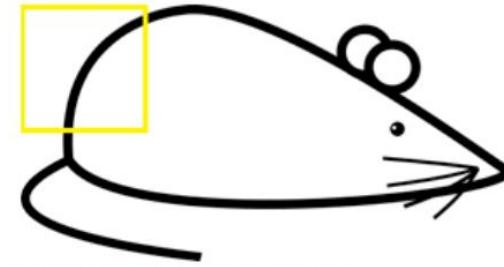
Сверточный слой



- Будем использовать не один, а F фильтров.
- Сверточный слой принимает на вход тензор $W \times H \times D$ и производит свертку набором из F фильтров $K \times K \times D$. Каждый фильтр дает двумерную матрицу активации, следовательно на выходе получается тензор $W \times H \times F$.
- Каждый фильтр ищет в окрестности своего пикселя некоторый паттерн.



Original image



Visualization of the filter on the image

Помните, всё что нам нужно, это умножить значения фильтра на исходные значения пикселей изображения.



Visualization of the receptive field

0	0	0	0	0	0	30	0
0	0	0	0	50	50	50	0
0	0	0	20	50	0	0	0
0	0	0	50	50	0	0	0
0	0	0	50	50	0	0	0
0	0	0	50	50	0	0	0
0	0	0	50	50	0	0	0
0	0	0	50	50	0	0	0

Pixel representation of the receptive field

*

0	0	0	0	0	0	30	0
0	0	0	0	30	0	0	0
0	0	0	30	0	0	0	0
0	0	0	30	0	0	0	0
0	0	0	30	0	0	0	0
0	0	0	30	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Pixel representation of filter

Multiplication and Summation = $(50*30)+(50*30)+(50*30)+(20*30)+(50*30) = 6600$ (A large number!)



Visualization of the filter on the image

0	0	0	0	0	0	0	0
0	40	0	0	0	0	0	0
40	0	40	0	0	0	0	0
40	20	0	0	0	0	0	0
0	50	0	0	0	0	0	0
0	0	50	0	0	0	0	0
25	25	0	50	0	0	0	0
0	0	0	0	0	0	0	0

Pixel representation of receptive field

*

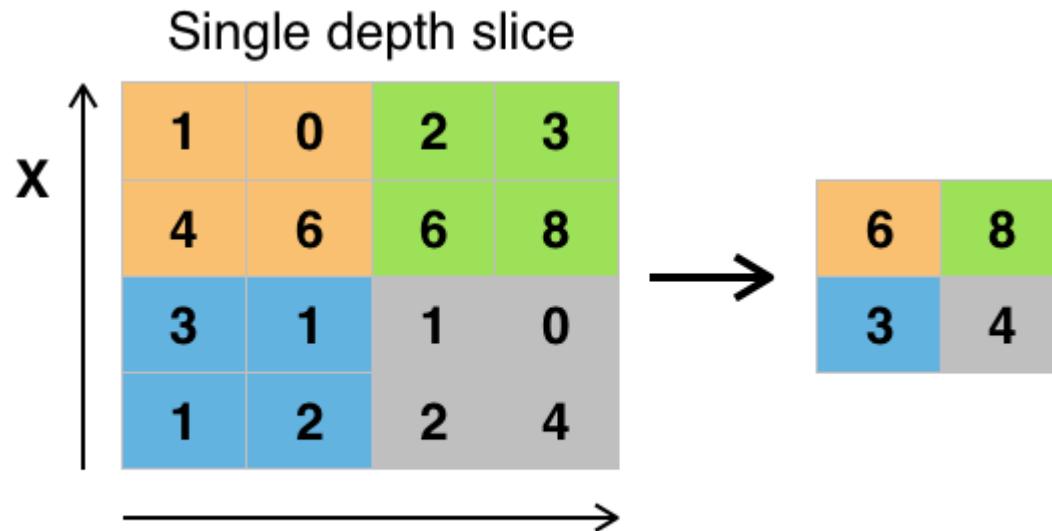
0	0	0	0	0	0	30	0
0	0	0	0	30	0	0	0
0	0	0	30	0	0	0	0
0	0	0	30	0	0	0	0
0	0	0	30	0	0	0	0
0	0	0	30	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Pixel representation of filter

Multiplication and Summation = 0



Субдискритизация (pooling)

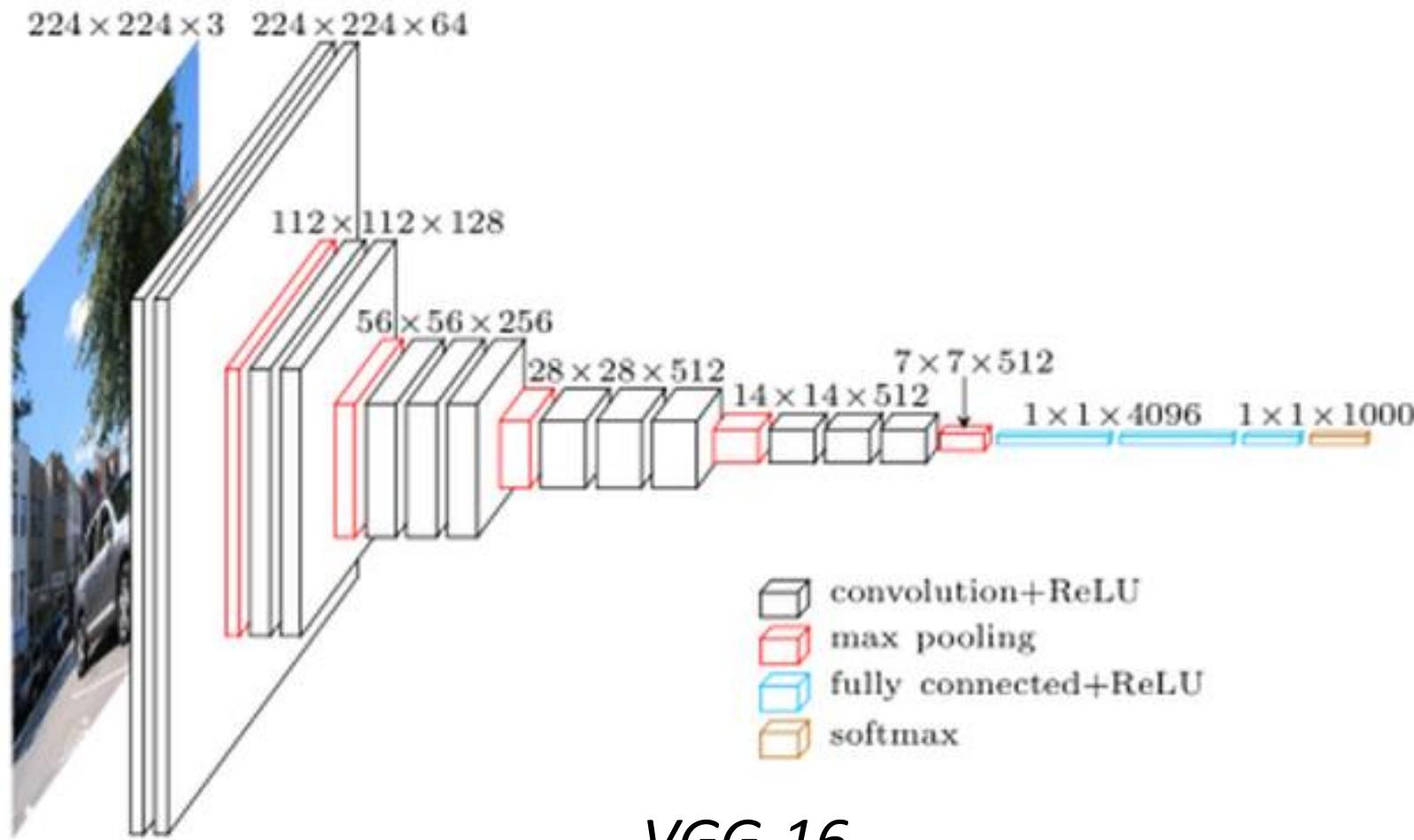


Изображение делится на регионы (напр. квадраты 2×2), и каждый регион заменяется на максимальное значение в этом регионе.

- Вырабатывается инвариатность к небольшим сдвигам
- Увеличение рецептивной области
- Уменьшение вычислительных затрат



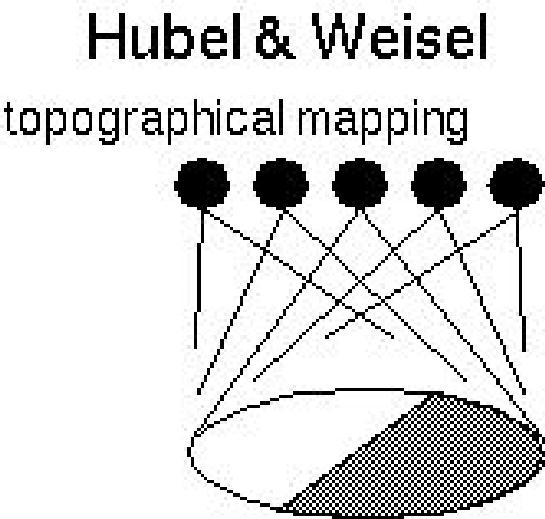
Примеры



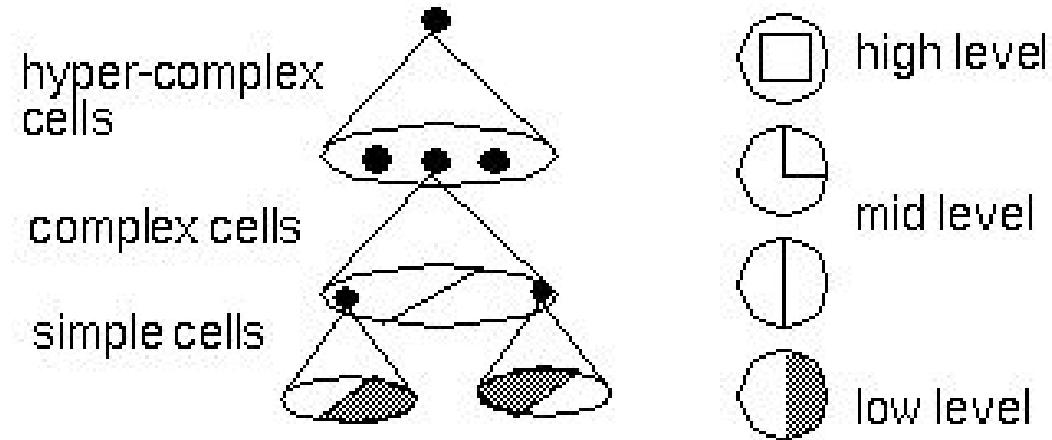
VGG-16



Понимание работы CNN



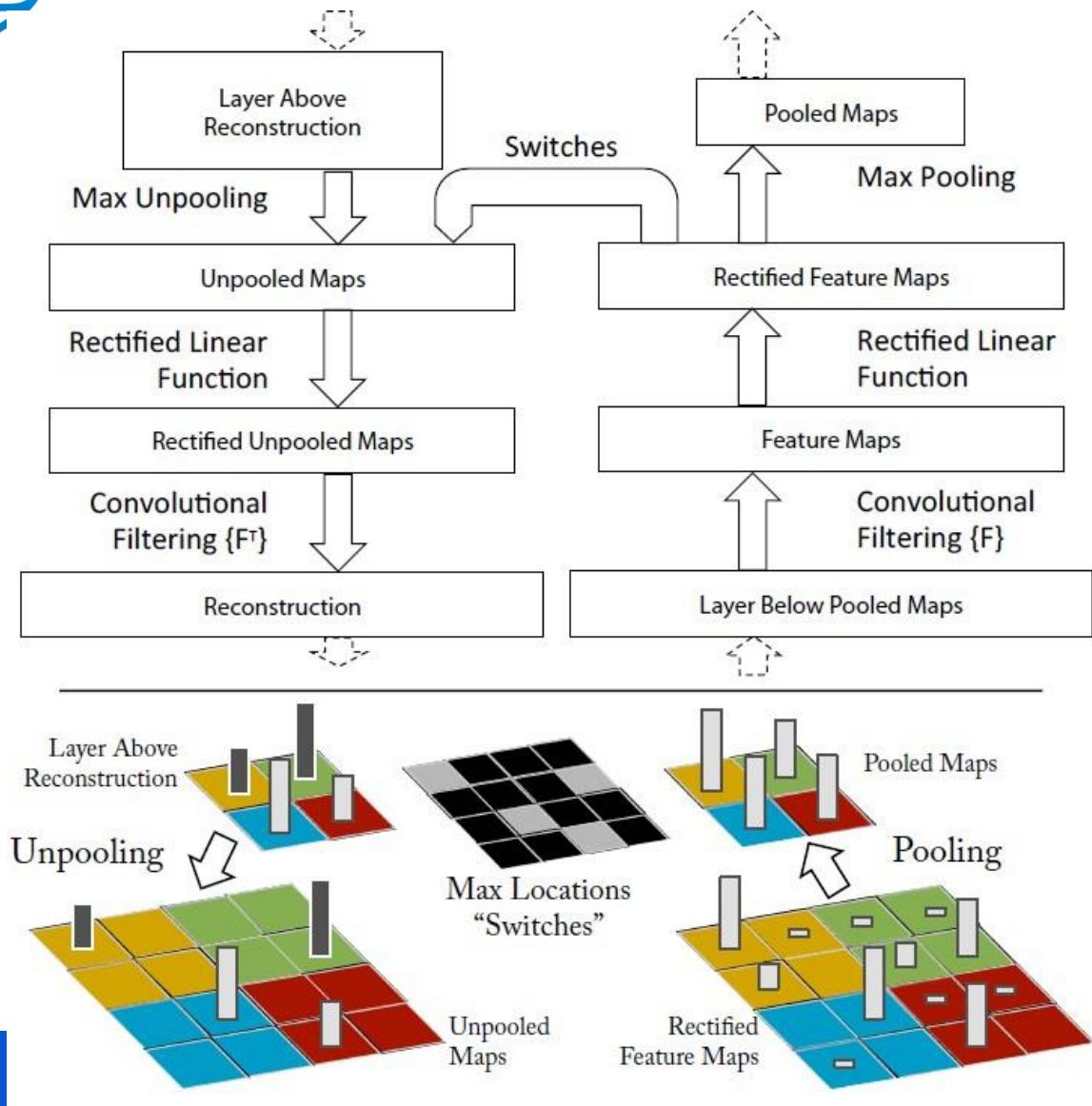
featural hierarchy



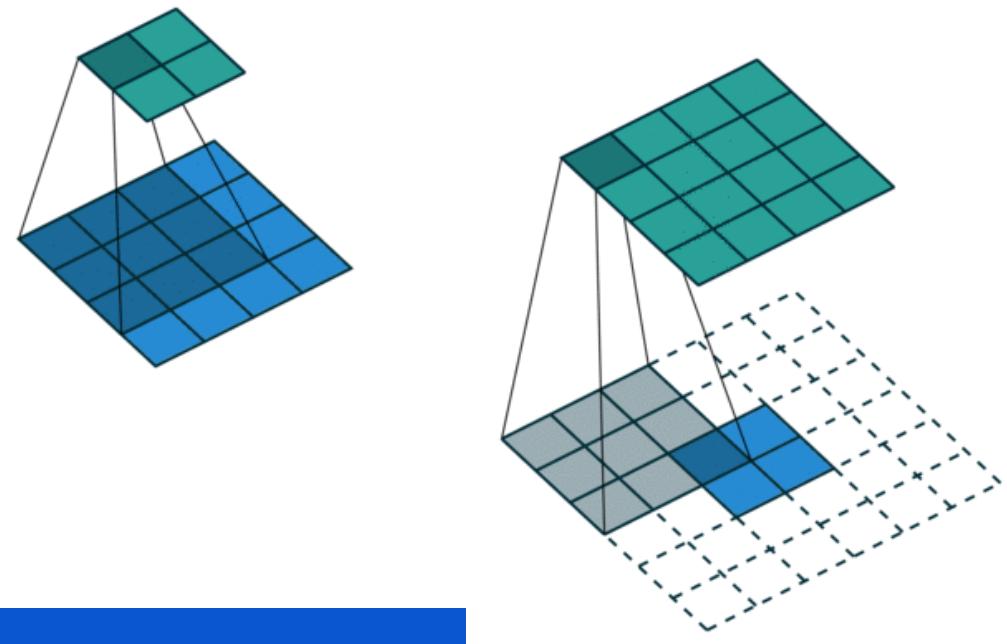
Показано, что мозг обрабатывает визуальную информацию иерархически: сначала находят границы, углы, а на более глубоких слоях – сложные объекты.

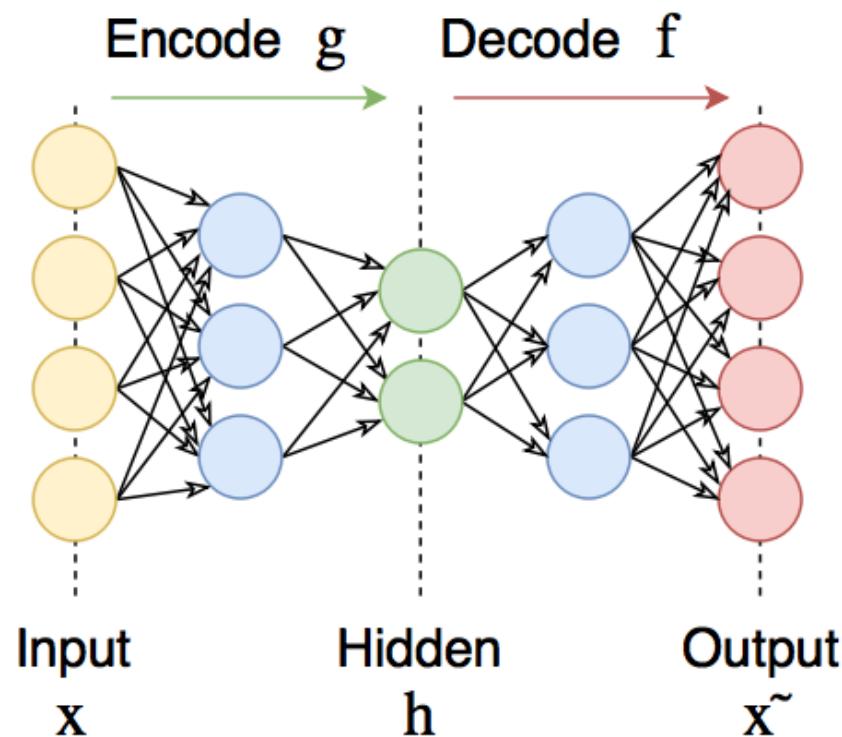


Deconvolutional network



– это сеть, которая интерпретирует CNN, т.е. показывает, какие пиксели **повлияли на активацию тех или иных выходов.**



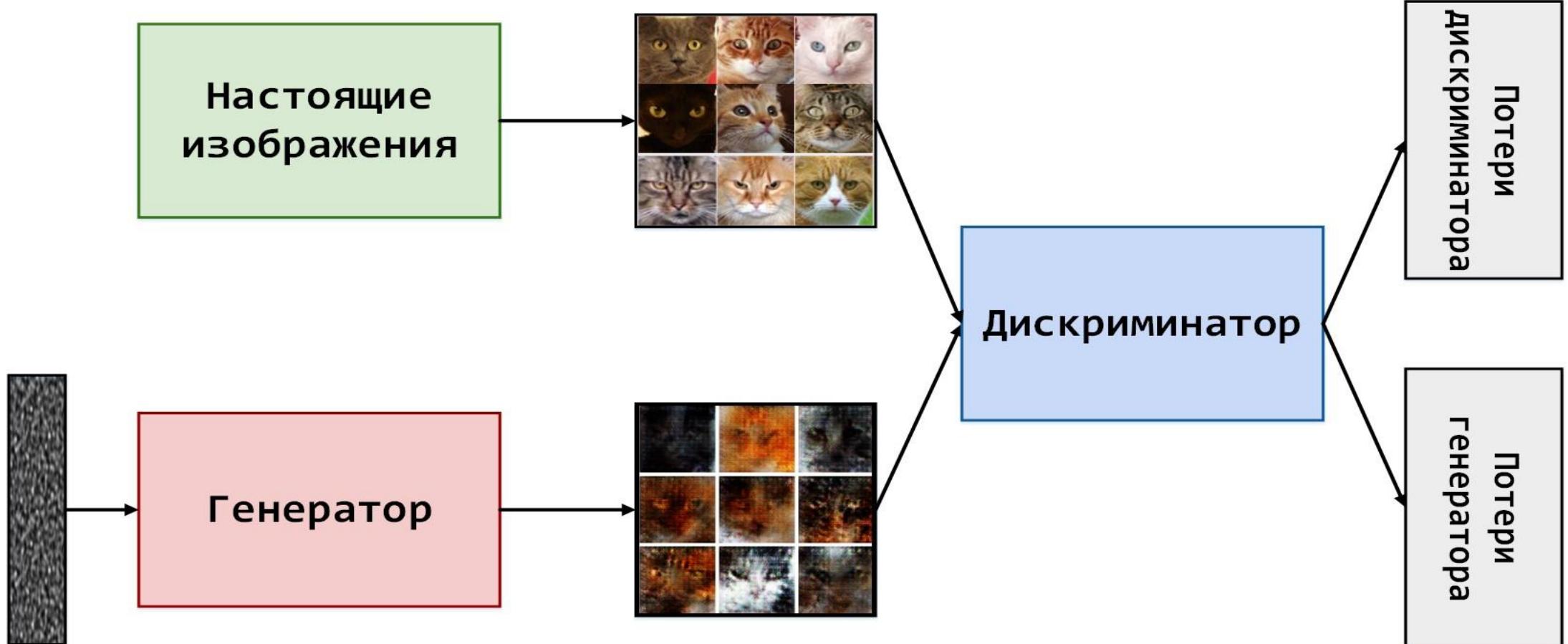


$$L(x, f(g(x))) \rightarrow \min$$

- Автоэнкодер – это специальная архитектура нейросети, состоящая из кодировщика и декодировщика. На месте их стыка образуется «бутылочное горлышко», на котором собираются наиболее важные признаки.
- Автоэнкодер пытается выучить тождественное преобразование, т.е. минимизировать разницу между входом и выходом сети.



Генеративно-состязательные нейронные сети





Обучение генеративно-состязательной сети

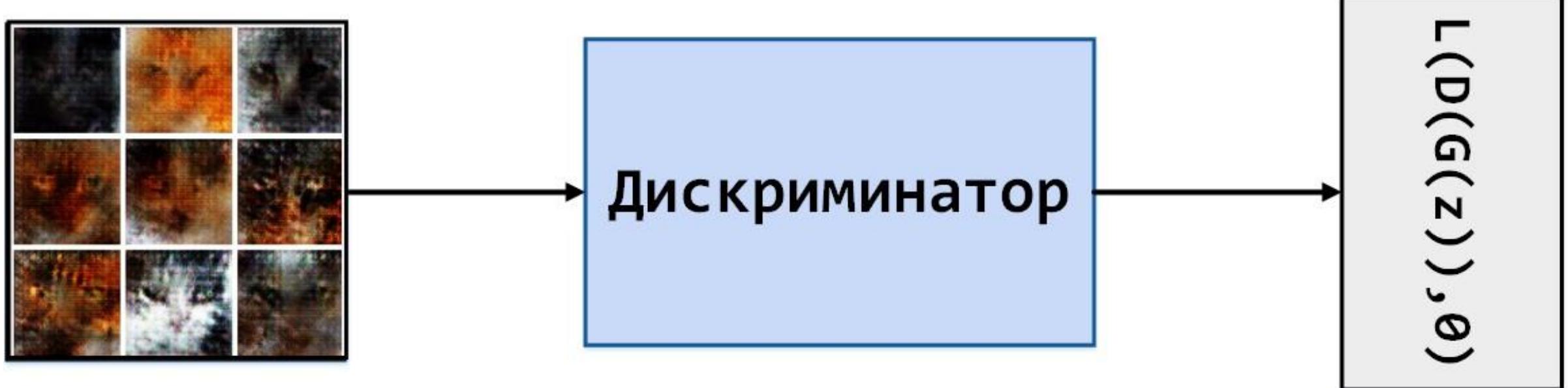
Обучаем дискриминатор предсказывать метку «1» на батче из реальных изображений.





Обучение генеративно-состязательной сети

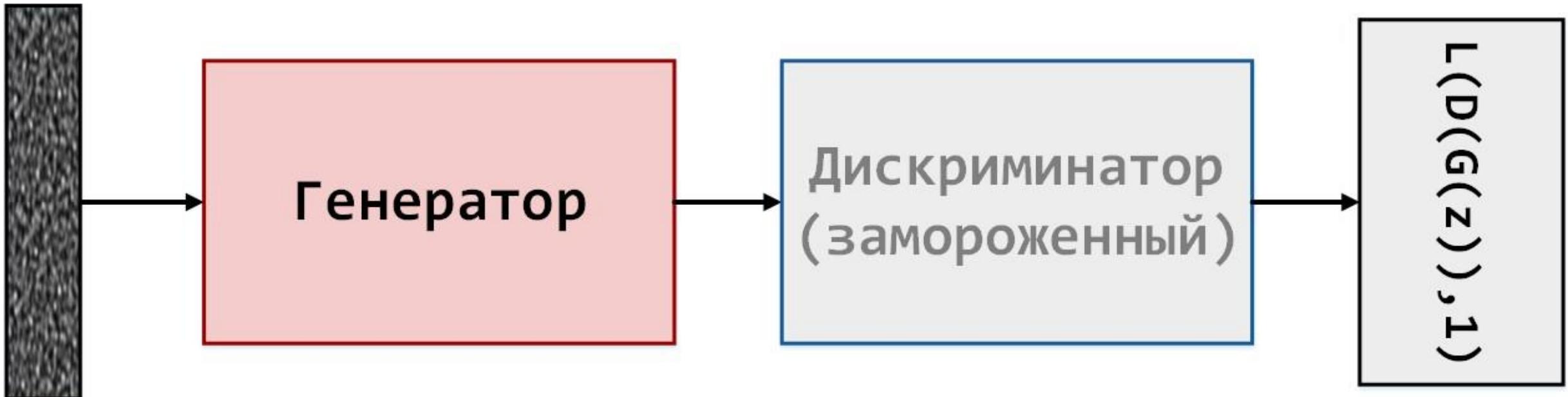
Обучаем дискриминатор предсказывать метку «0» на батче из изображений, созданных генератором.





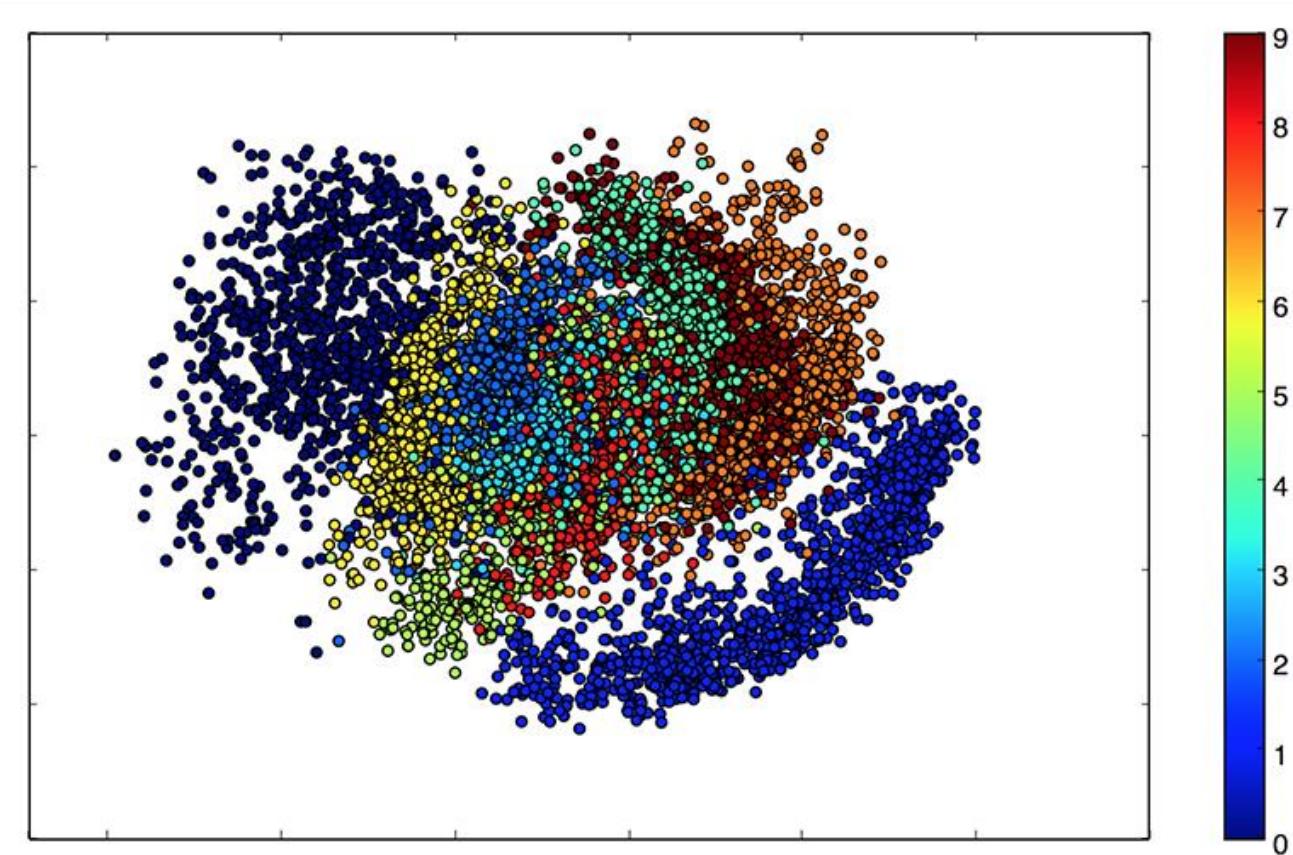
Обучение генеративно-состязательной сети

Замораживаем весовые коэффициенты дискриминатора и обучаем полную сеть предсказывать метку «1» для произвольного входного вектора.





Скрытое пространство

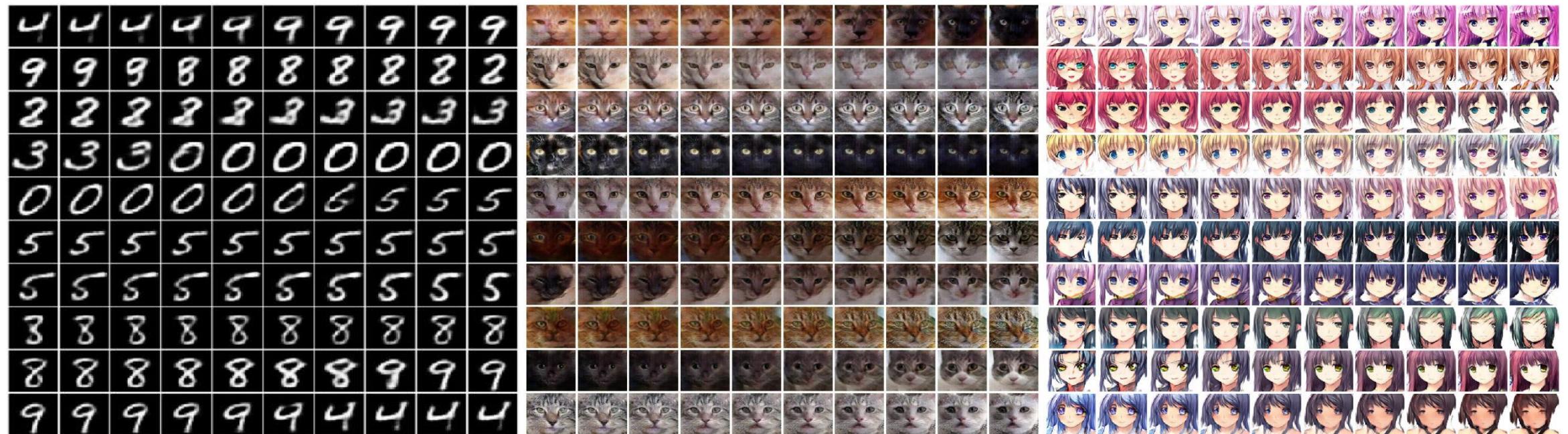


Скрытое пространство – маломерное пространство, в которое кодировщик отображает данные. Его визуализация позволяет получать проекции, лучшие чем какой-либо другой классический метод.



Интерполяция в скрытом пространстве

Выбирая два вектора (z_1 и z_2) в скрытом пространстве и выполняя линейную интерполяцию между ними, можно получить интерполяцию в пространстве изображений: $z = t \cdot z_1 + (1 - t) \cdot z_2$, где t – число $\in [0, 1]$





Ошибки при обучении

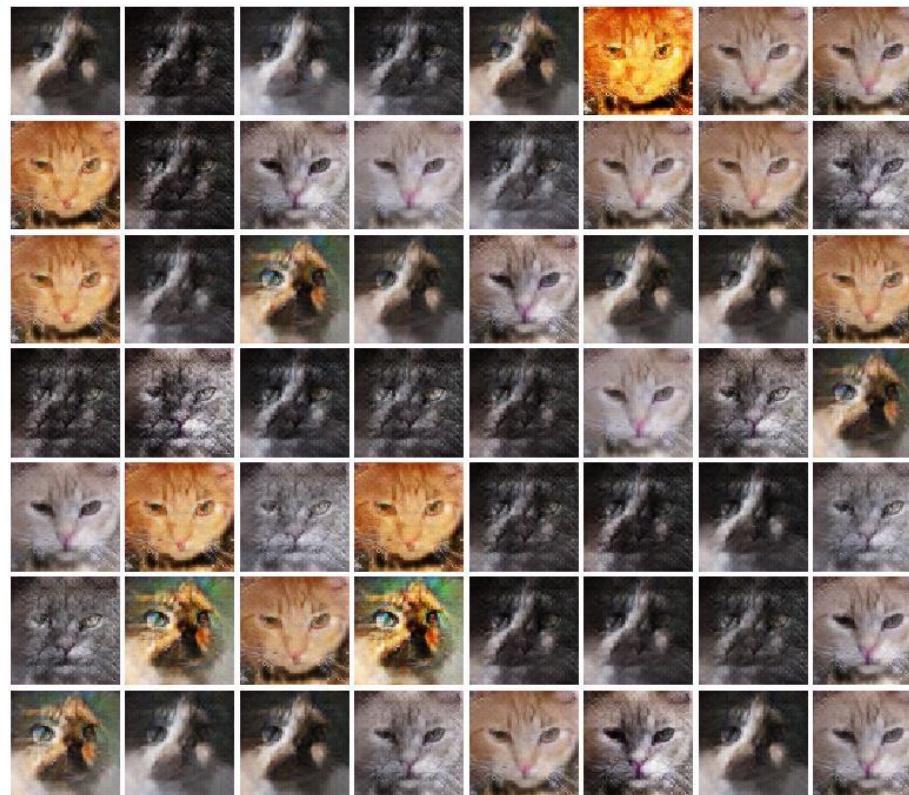
Затухающие градиенты – если дискриминатор слишком хорош, то обучение генератора может не сработать из-за затухающих градиентов. Фактически, оптимальный дискриминатор не предоставляет достаточно информации для работы генератора.

Неспособность сойтись. GAN'ы сложно обучать. Многие генеративно-состязательные модели вовсе не способны сойтись в приемлемую точку.



Ошибки при обучении

Коллапс моды – режим работы GAN, при котором генератор выдаёт лишь сильно ограниченное количество изображений.





Приёмы для стабильного обучения

- нормализация входных данных (изображения в диапазоне $[-1, 1]$)
- \tanh в качестве функции активации генератора;
- нормальное распределение вместо равномерного для шума;
- раздельные пакеты для настоящих и фейковых изображений;
- использование батч-нормализации;
- использование LeakyReLU вместо ReLU;
- добавление шума в метки ($[0, 0.3]$ и $[0.7, 1.2]$ вместо 0 и 1)
- использование Adam со значением момента 0.5
- использование свёрточных слоёв с шагом > 1 вместо макспулинга;



Super resolution GAN (пример)

bicubic
(21.59dB/0.6423)



SRResNet
(23.53dB/0.7832)



SRGAN
(21.15dB/0.6868)



original





CNN для распознавания звуков и текстов

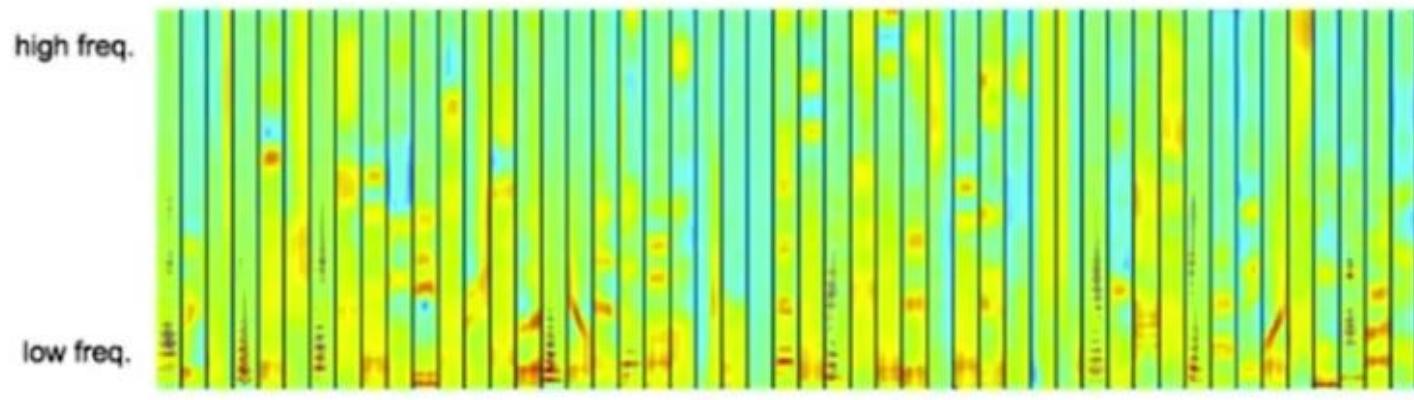
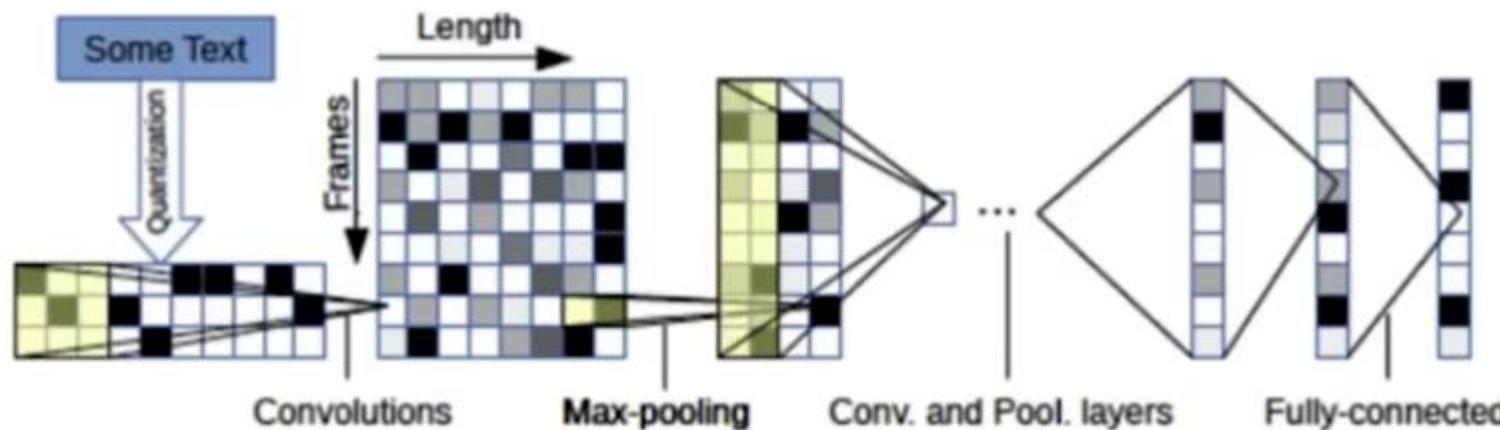


Рис.: Спектрограмма голосового сигнала





Рекуррентные нейронные сети

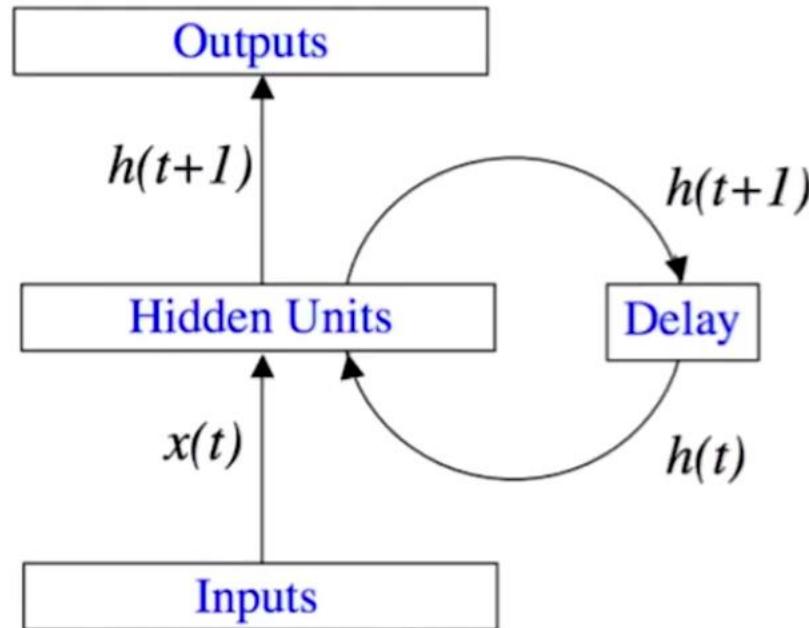


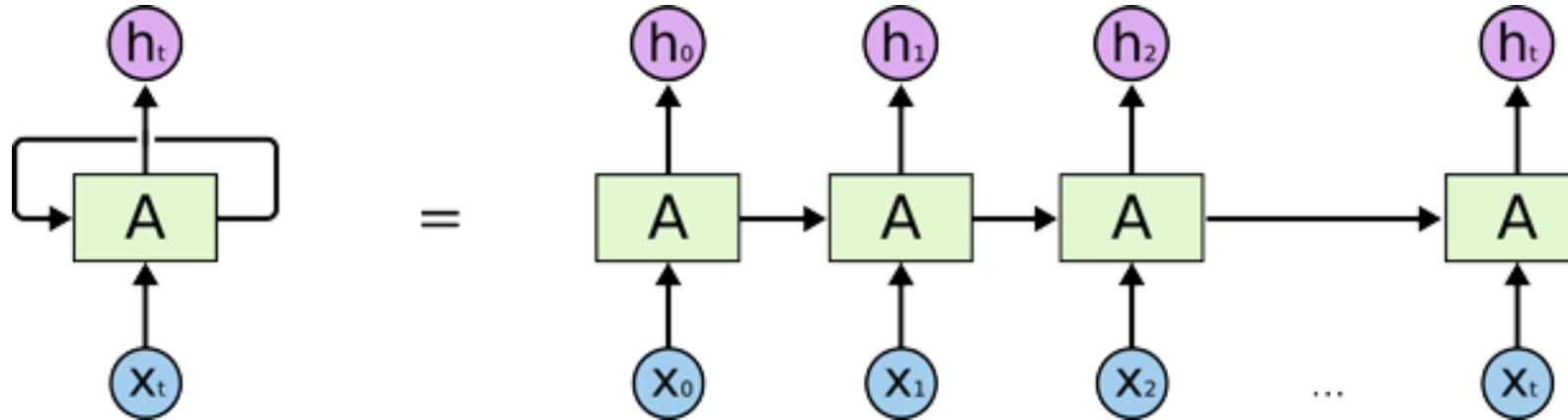
Рис.: RNN с задержкой на скрытом слое

- Рекуррентная нейронная сеть (RNN) — архитектура нейронных сетей, где связи между элементами образуют направленный цикл.
- Наличие таких циклов делает эту архитектуру идеальной для обработки последовательностей и данных, распределенных во времени.

- Все биологической нейронной сети – рекуррентные
- RNN моделирует динамическую систему
- Универсальная теорема аппроксимации говорит, что с помощью RNN можно смоделировать поведение любой динамической системы
- Существует много алгоритмов обучения RNN без явного лидера.



Общий случай



- В общем случае RNN может запоминать некоторый «контекст» на скрытых слоях
- Для этого она обучается методом обратного распространения ошибки развернутого во времени.



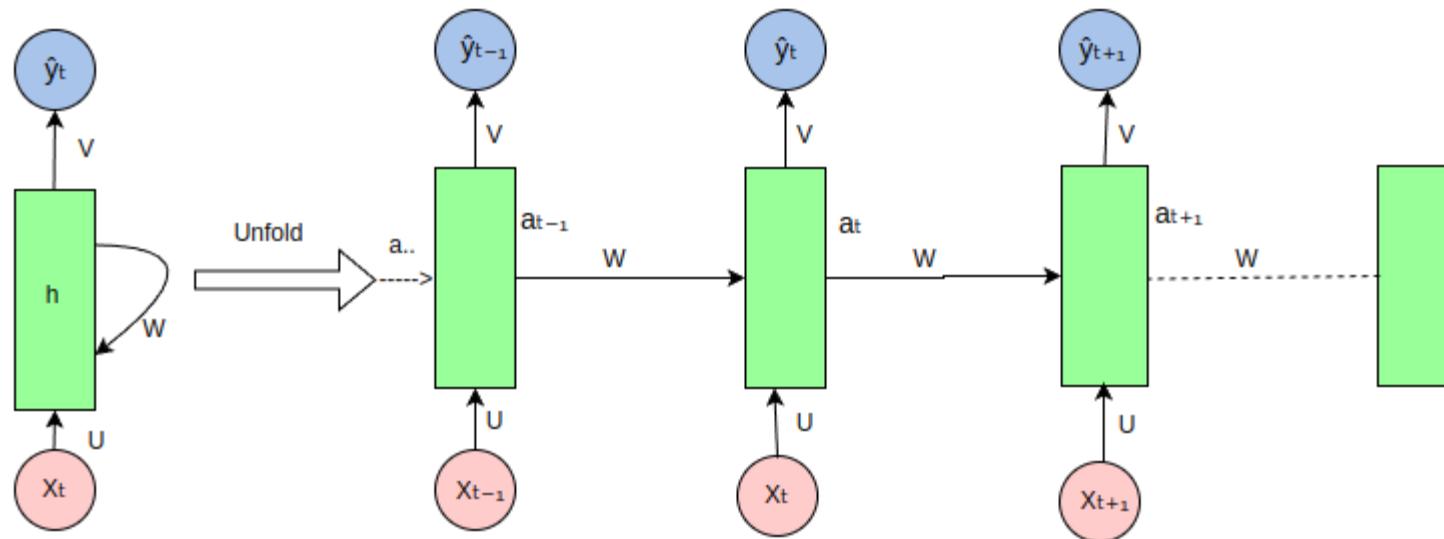
Применение RNN

- Прогнозирование временных рядов
- Моделирование последовательностей
 - преобразования (напр. из звука в текст)
 - предсказание следующего элемента последовательности (напр. следующего слова в предложении)
- Анализ контекста и внутренней структуры



Проблемы с RNN

В частности, LSTM (Long Short Term Memory) и GRU (Gated Recurrent Unit) были популярными RNN, способными кодировать богатую семантику слов в тексте. Они работают последовательно, обрабатывая по одному токену за раз и сохраняя «память» всех токенов, обработанных моделью.



Основные ограничения RNN:

- 1. Последовательная обработка:** невозможность параллелизации
- 2. Проблема исчезающих градиентов:** сложность обучения долгосрочных зависимостей
- 3. Ограниченнная память:** потеря информации при обработке длинных последовательностей
- 4. Вычислительная неэффективность:** медленное обучение на больших данных



Проблемы с CNN

- 1. Ограниченнное рецептивное поле:** сложность захвата долгосрочных зависимостей
- 2. Фиксированные фильтры:** неадаптивность к различным контекстам
- 3. Иерархическая обработка:** необходимость глубоких сетей для больших рецептивных полей
- 4. Локальность:** фокус на локальных паттернах, а не глобальных связях

Рецептивное поле в этих типах CNN зависит от размера их фильтров и количества используемых свёрточных слоёв. Увеличение значения этих гиперпараметров увеличивает сложность модели, что может привести к исчезновению градиентов или даже невозможности обучения моделей.



5. Трансформеры - революция в архитектуре

Трансформеры — это модель глубокого обучения, использующая механизм самовнимания (self-attention), который по-разному взвешивает значимость каждой части входных данных. Они используются в основном в области обработки естественного языка (Natural Language Processing — NLP) и компьютерного зрения (Computer Vision — CV).

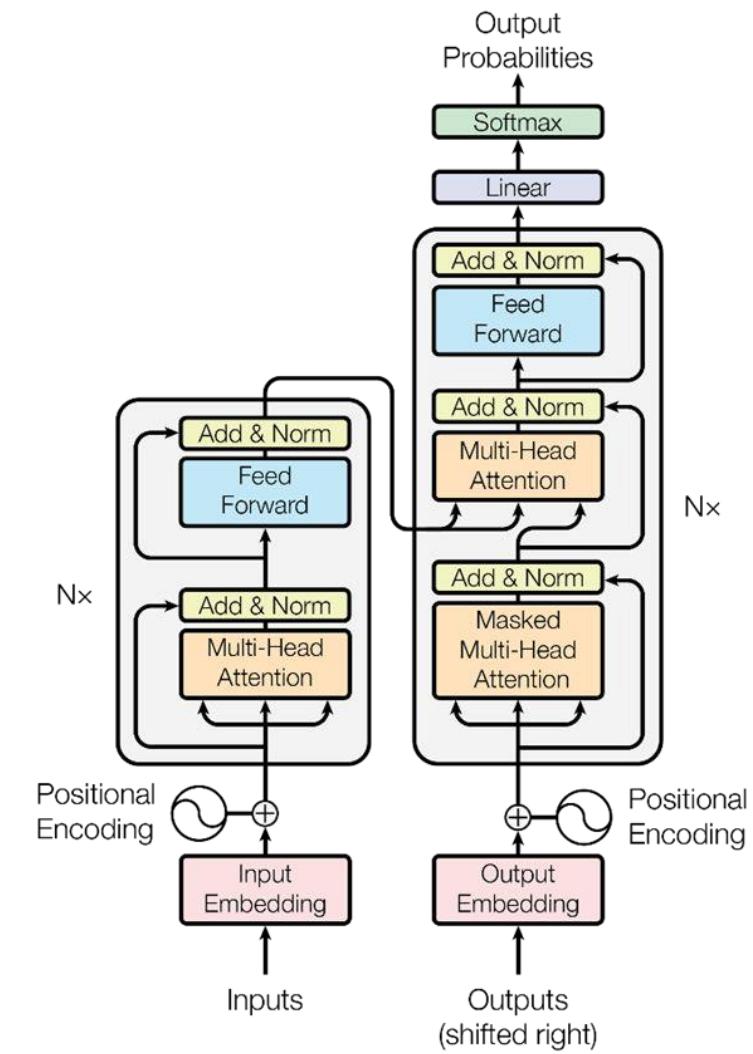
Трансформеры, как и RNN, предназначены для обработки последовательных входных данных (например, естественного языка) с применением к таким задачам, как перевод и суммирование текста. Однако, в отличие от RNN, трансформеры обрабатывают весь вход одновременно. Механизм внимания определяет контекст для любой позиции во входной последовательности.

Ключевые преимущества трансформеров:

- **Параллелизация:** обработка всех элементов последовательности одновременно
- **Долгосрочные зависимости:** эффективное моделирование связей между удалёнными элементами
- **Масштабируемость:** возможность обучения на больших объемах данных
- **Универсальность:** применимость к различным типам данных

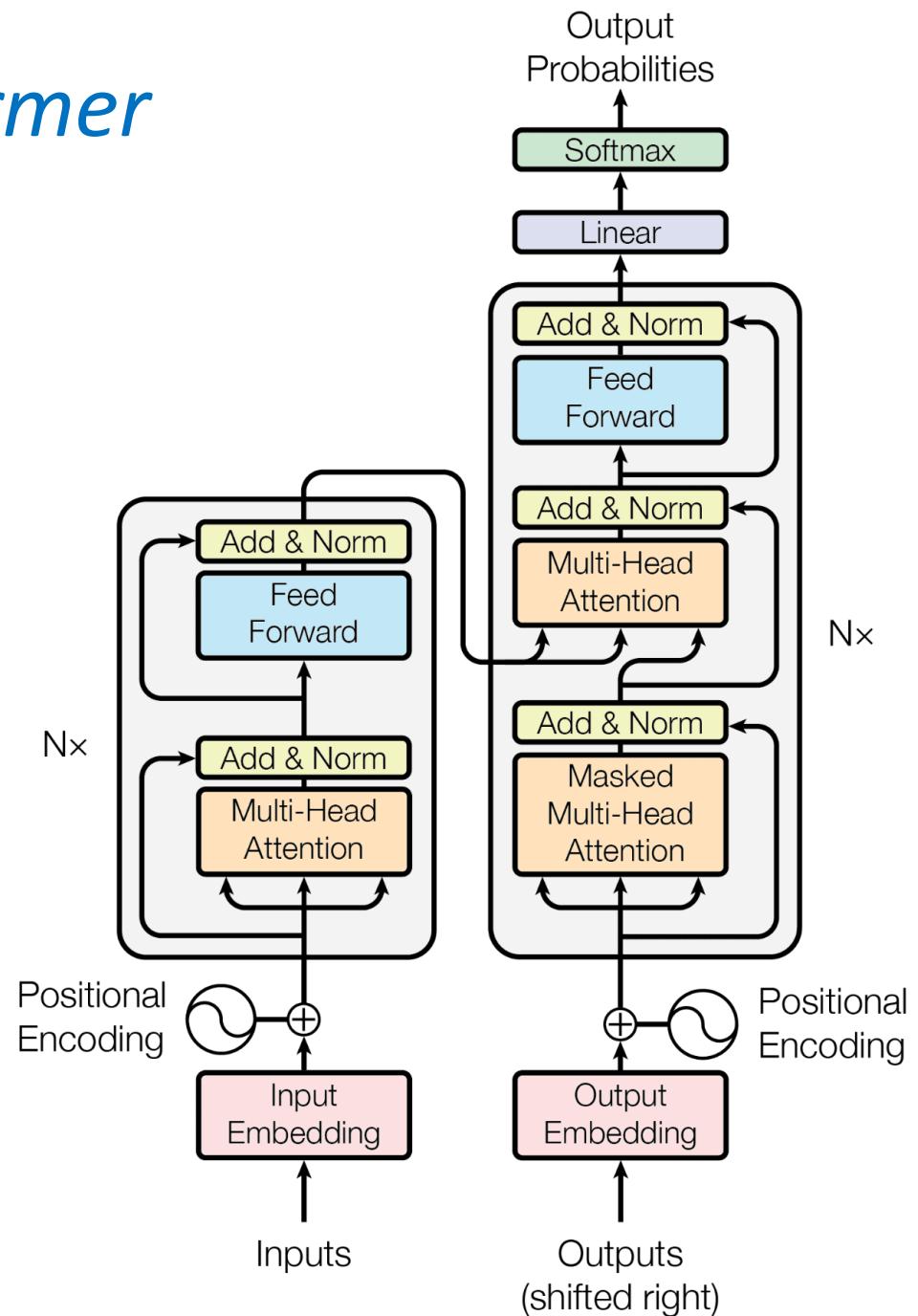
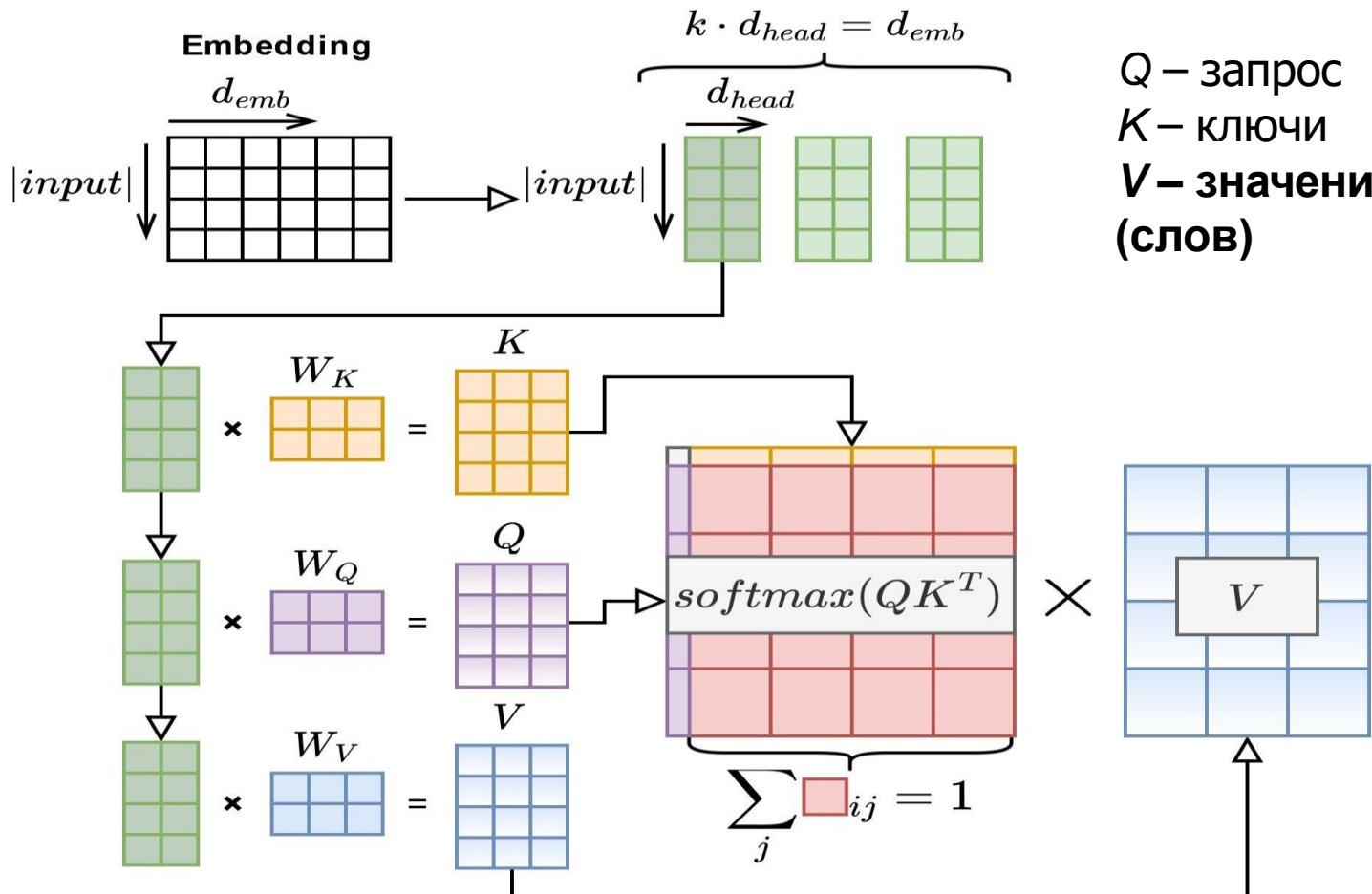
Transformer (2017)

- Токенизация слов
- Позиционное кодирование
- Преобразование векторов через трансформер-блоки
 - Multi-head attention (МНА)
 - Feed-forward (FFN)
 - LayerNorm
- Итоговое предсказание слова линейным слоем (lm head)





Общая архитектура Transformer



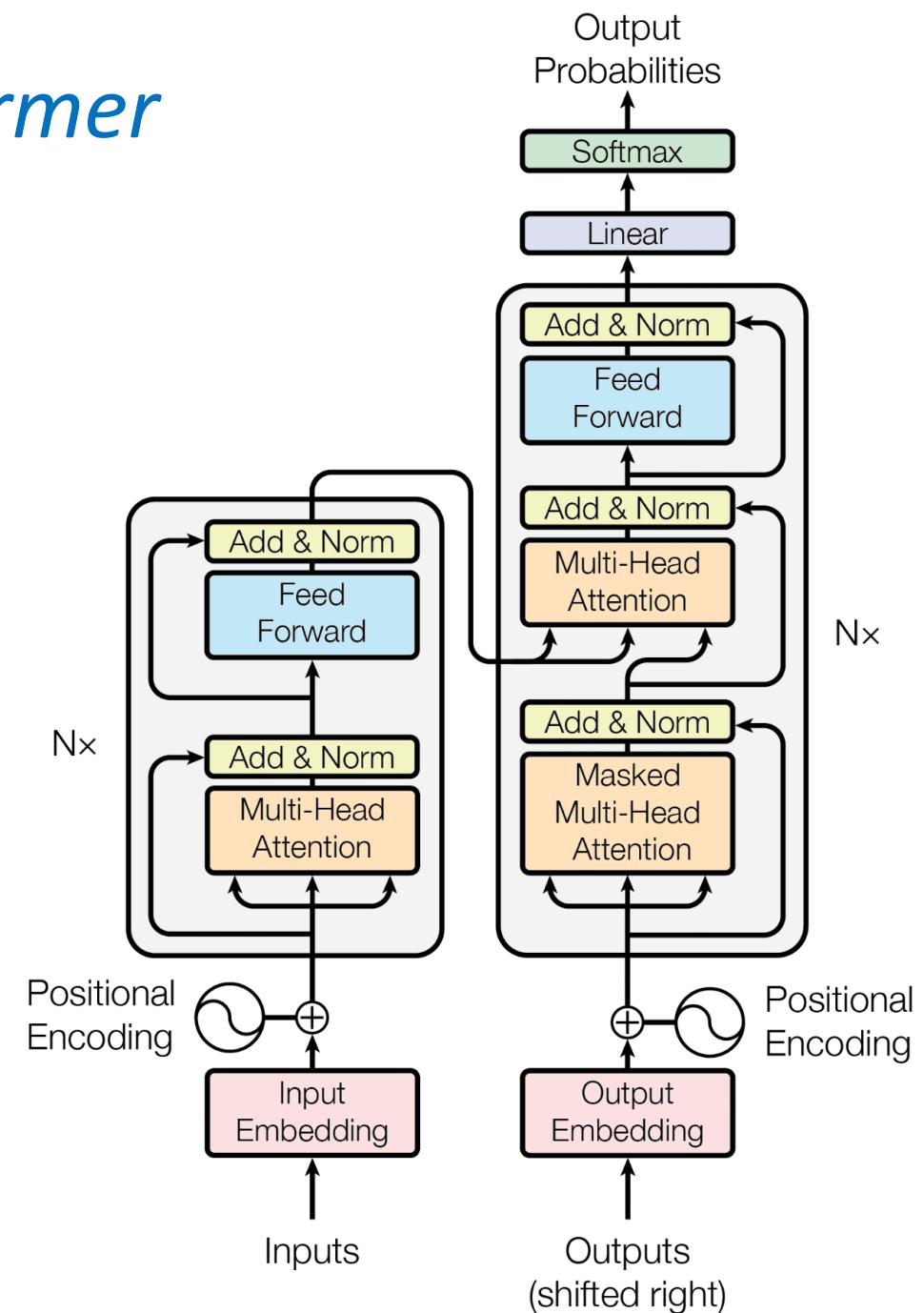


Общая архитектура Transformer

$$\begin{array}{l} \mathbf{x} \times \mathbf{W}^Q = \mathbf{Q} \\ \mathbf{x} \times \mathbf{W}^K = \mathbf{K} \\ \mathbf{x} \times \mathbf{W}^V = \mathbf{V} \end{array}$$

$\text{softmax}\left(\frac{\mathbf{Q} \times \mathbf{K}^T}{\sqrt{d_k}}\right) = \mathbf{Z}$

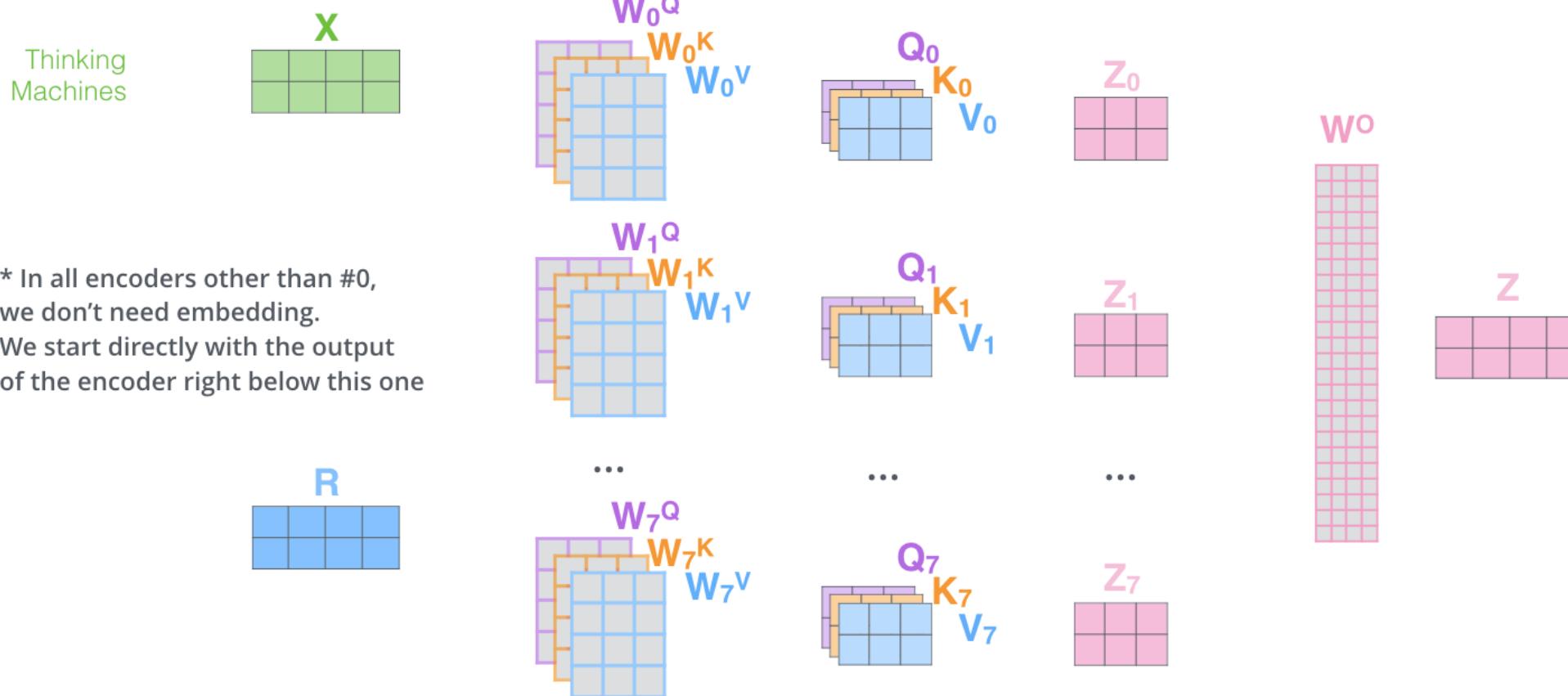
\mathbf{V}





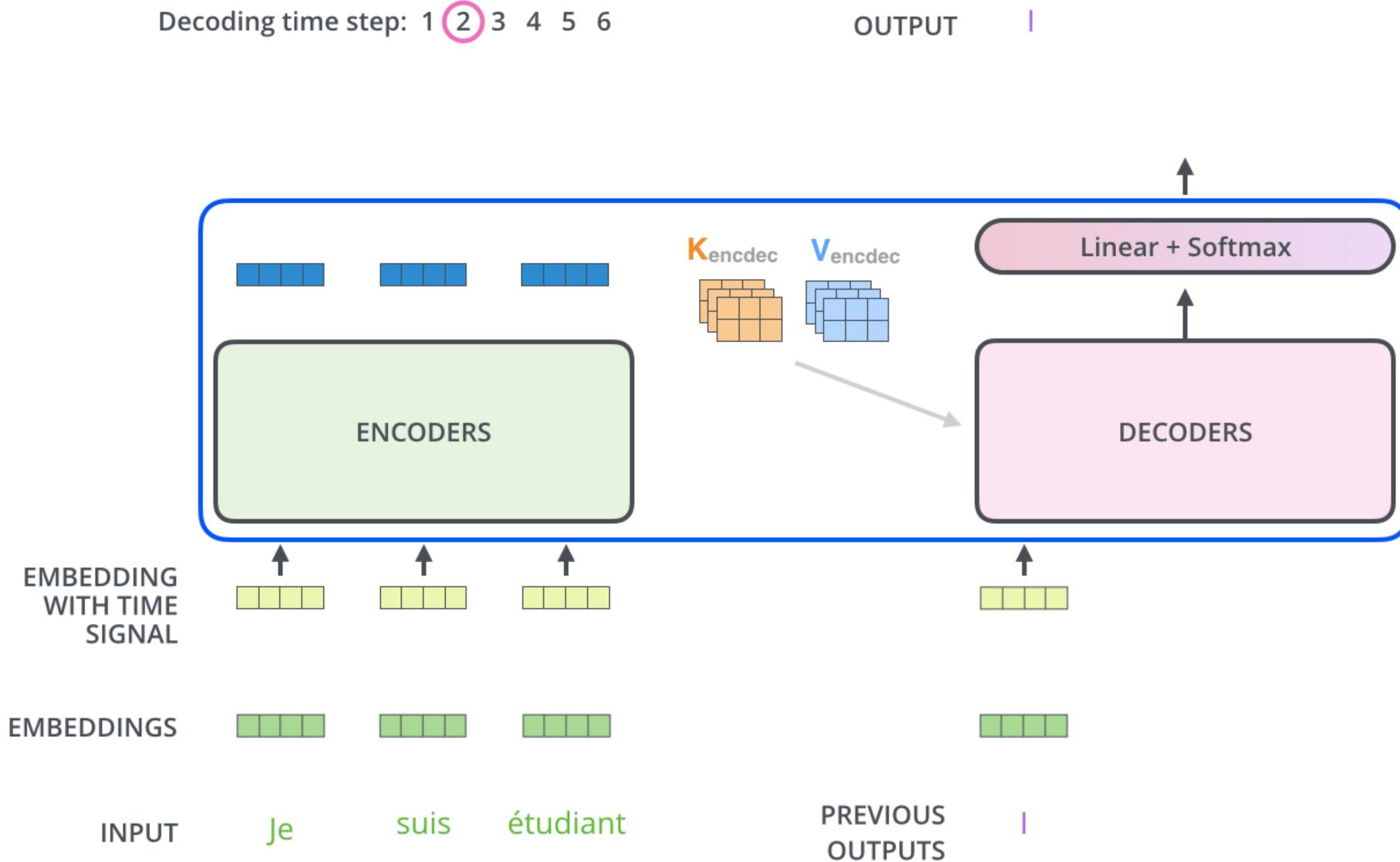
Общая архитектура Transformer

- 1) This is our input sentence* X
- 2) We embed each word*
- 3) Split into 8 heads. We multiply X or R with weight matrices
- 4) Calculate attention using the resulting $Q/K/V$ matrices
- 5) Concatenate the resulting Z matrices, then multiply with weight matrix W^o to produce the output of the layer



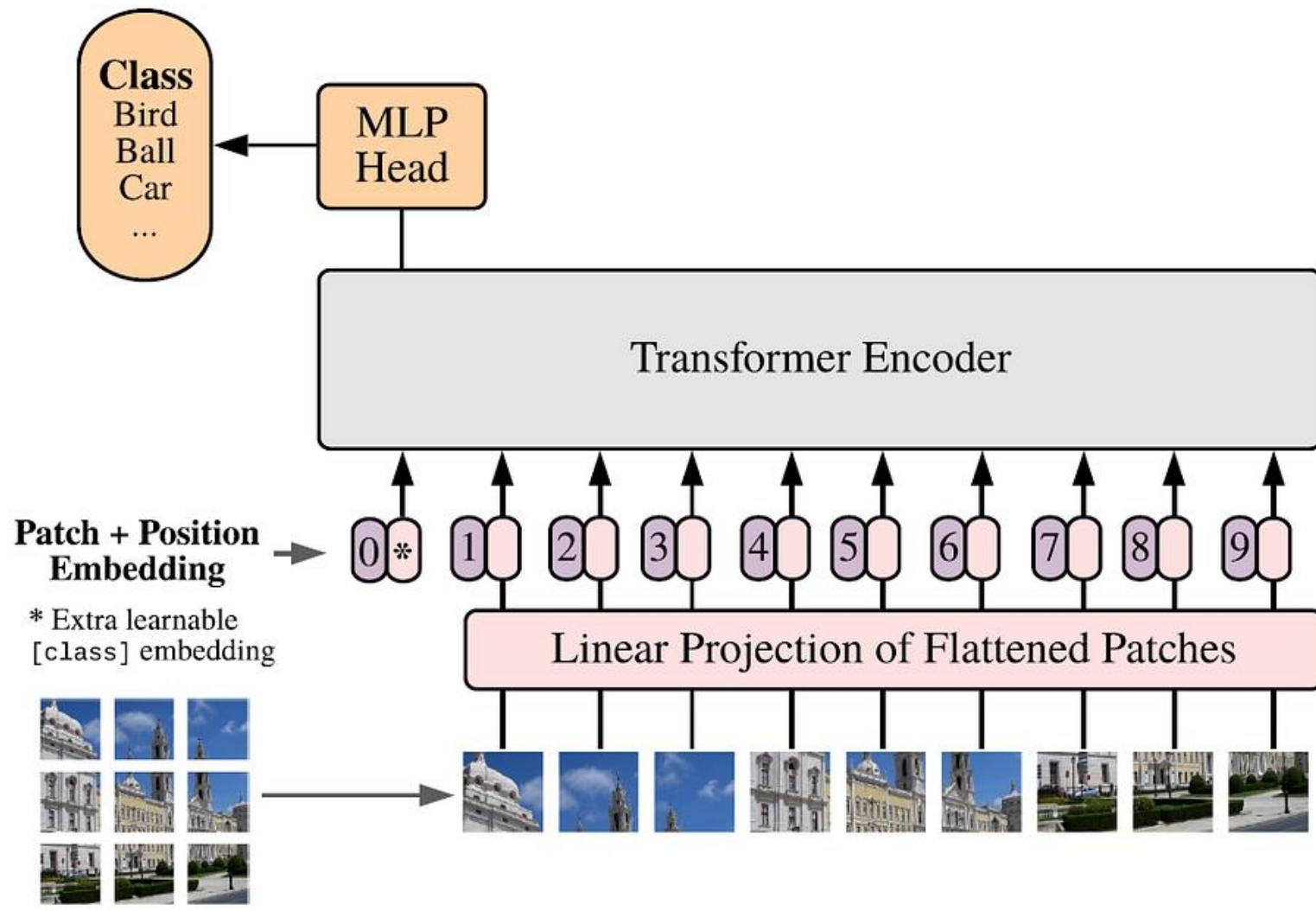


Общая архитектура Transformer

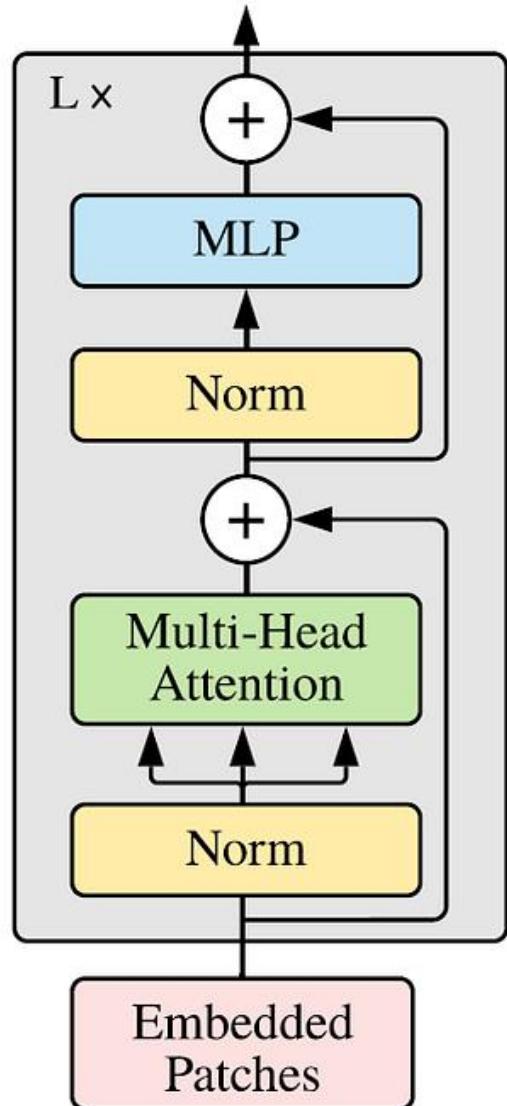




Vision Transformer (ViT)



Transformer Encoder





6. Эра больших языковых моделей - от BERT к GPT

Эффективность трансформеров

В NLP цель нейроязыковых моделей состоит в том, чтобы создать векторные представления (embeddings), которые кодируют как можно больше информации о семантике слова в тексте. Эта семантика не ограничивается определением слова — на самом деле многие слова сами по себе бессмысленны, если мы не знаем контекста, к которому они принадлежат.

Пример контекстной зависимости:

В предложении «Трансформеры эффективны, потому что они быстрые» векторное представление слова «они» будет бессмысленным, если не учитывать, что оно относится к «трансформерам».

Оптимальные модели должны уметь отображать эти зависимости между словами даже при работе с большими текстами, где эти слова могут быть удалены друг от друга на значительное расстояние. Модель с такой способностью может кодировать долгосрочные зависимости. Трансформеры способны находить эти зависимости между словами эффективно.



Рост популярности LLM в мире

Оценка стоимости LLM компаний инвесторами

- Mistral - 5.8 миллиарда
- XAI - 24 миллиарда
- Anthropic - 40 миллиардов
- OpenAI - 157 миллиардов

Для сравнения (market cap):

- Siemens ~150 миллиардов
- Nvidia ~3 триллиона (рост **x10** за 4 года)



Классификация моделей

LLM могут иметь различную архитектуру в зависимости от их целей, вычислительных ресурсов и задач обучения. Также их можно разделить по архитектурам:

- **(На базе энкодера) BERT** (Bidirectional Encoder Representations from Transformers) — модель, которая смотрит на текст в обоих направлениях для лучшего понимания контекста.
- **(На базе декодера) GPT** (Generative Pre-trained Transformer) — модель, предназначенная в первую очередь для генерации текста.
- **(Энкодер + декодер) T5** (Text-to-Text Transfer Transformer) — модель, которая рассматривает все задачи обработки естественного языка как преобразование текста в текст.
- **MoE** (Mixture of Experts) — модель, которая состоит из нескольких экспертов (предобученных моделей) для выполнения определённой задачи.



OpenAI GPT-1 (2018)

- 12 слоев **Transformer decoder** (~117 млн.),

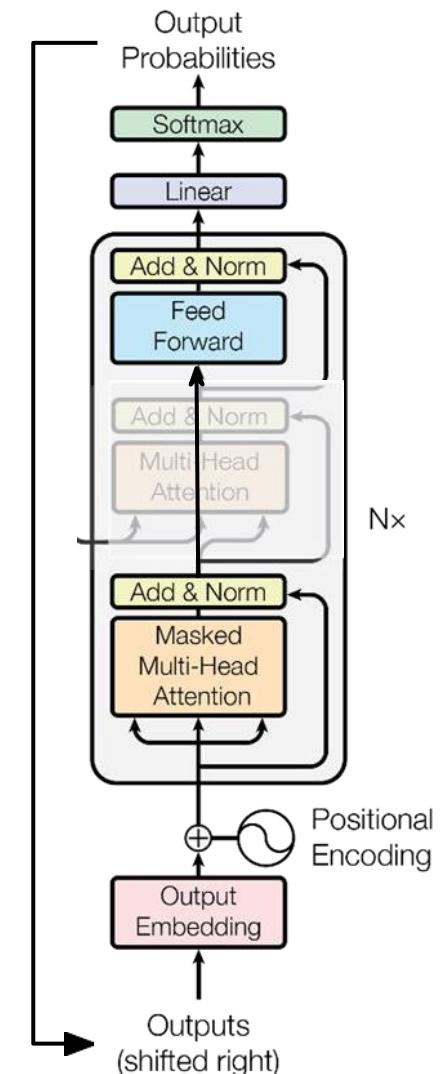
- Обучение в **2 этапа**:

- Предобучение (pre-training) на задаче **моделирования языка**

$$\max_{\Theta} \sum_{0 \leq i \leq n} \log P(w_i | w_{i-1} \dots w_0; \Theta)$$

w - слова последовательности, Θ - параметры модели

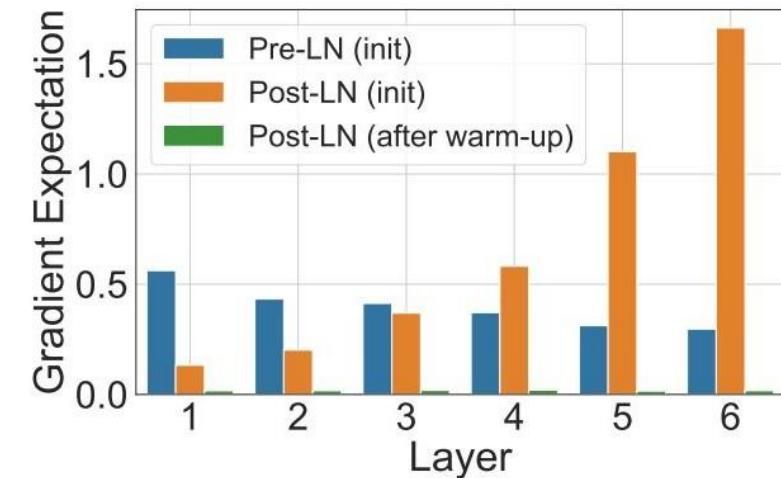
- Дообучение (fine-tuning) на целевые задачи
- Предобучался только на художественной литературе





GPT-2 (2019) – универсальный генератор текстов

- **Улучшенная архитектура:**
предварительная нормализация (**Pre-LN**)
входных данных для стабилизации градиентов
- **Больше параметров:**
в 4 раза больше слоев (**1.5 млрд параметров**)
– больше потенциальных знаний (capacity)
- **Новая парадигма:** любой текст содержит **подсказки к генерации (prompt)**
и обучаясь на большом наборе текстов модель учится их понимать





GPT-3 (2020) – первая коммерческая модель

- **Ориентация на рынок:** модель как облачный сервис
- **175 млрд параметров:** 96 слоев Transformer-decoder
- **Оптимизация потребления памяти:** половина слоев внимания используют разреженные матрицы (локальные окна)
- **Развитие парадигмы подводок (prompt):**
“обучение в контексте” (in-context learning)
- **Обучение на доверенных данных:** примеры для обучения смешиваются пропорционально их качеству (согласно экспертам)
- **В 15 раз больше данных:** добавлена очищенная коллекция CommonCrawl (570GB) и два новых корпуса книг (95GB)



Foundation vs Instruct

- LLM условно можно разделить на 2 вида:
 - **Базовые модели**, foundation models, которые обучались предсказывать следующее слово на просто текстах. Результат процедуры пре-трейна.
 - **Инструктивные модели** - являются дообученными базовыми моделями на инструктивных данных.
- Качество инструктивных моделей зависит от:
 - Качества базовой модели,
 - Инструктивного датасета,
 - Процедуры дообучения на инструктивном датасете.



Промпting

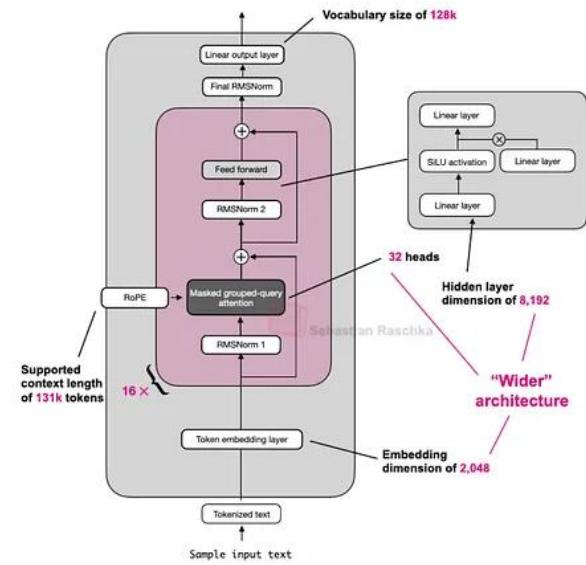
- Имеется LLM и некоторая задача, что делать дальше?
- В первую очередь, оценка качества модели “как есть”: составить различные промпты подходящие под задачу, протестировать их качество.
 - Необходимо составлять хорошие промпты не только для instruct моделей.
- Если zero-shot/few-shot/rag и тп не устраивает по качеству, то тогда можно думать про дообучение.



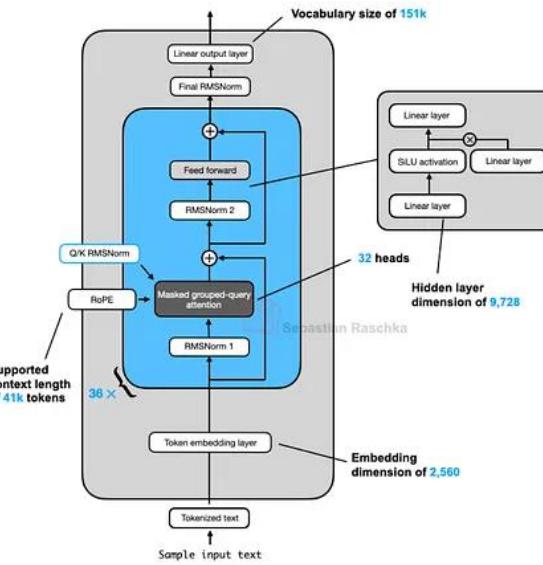
- Популярность LLM продолжает расти
- В основе современных LLM лежит **архитектура трансформер и механизм внимания**
- Развитие LLM прямо связано с **вычислительными ресурсами**
- Хорошая LLM = **Данные + GPU + специалисты**
- Важно уметь **правильно использовать LLM** для достижения результата⁴



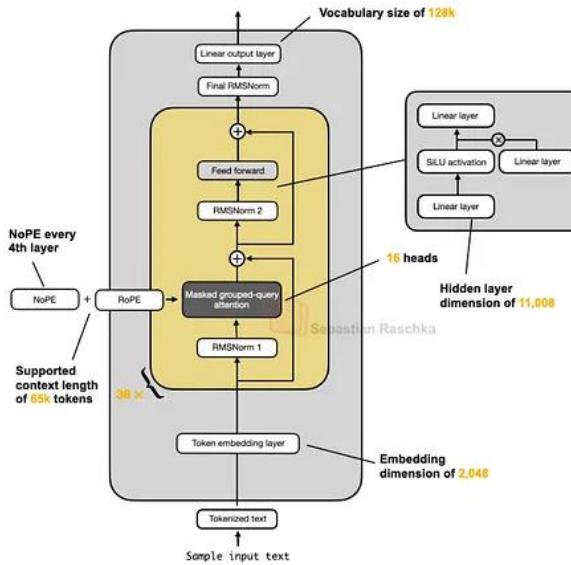
Llama 3.2 1B



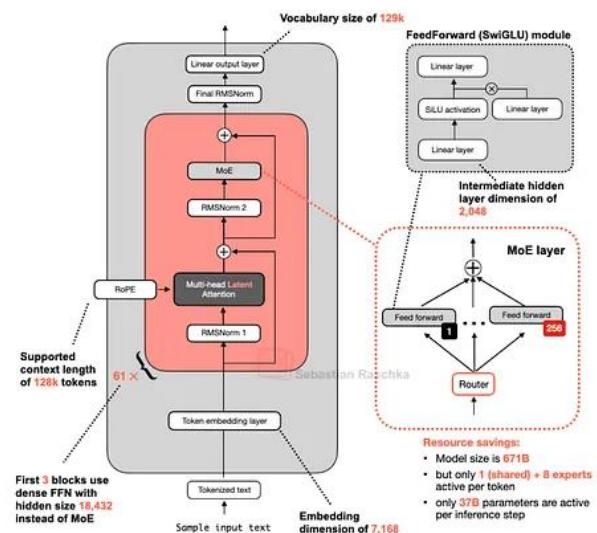
Qwen3 4B



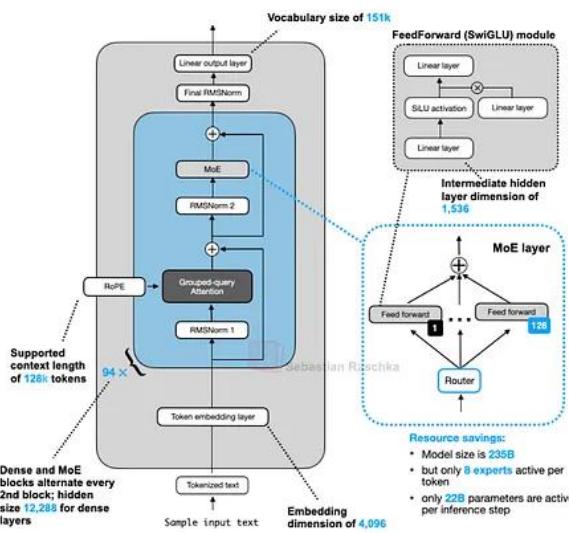
SmollM3 3B



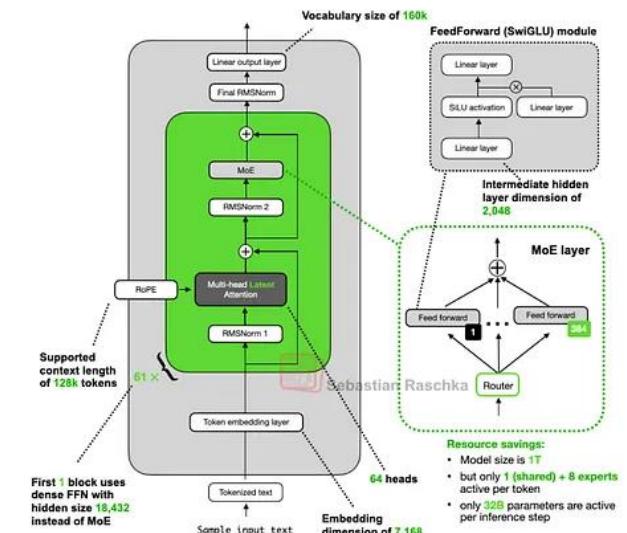
DeepSeek V3 (671B)



Qwen3 235B-A22B



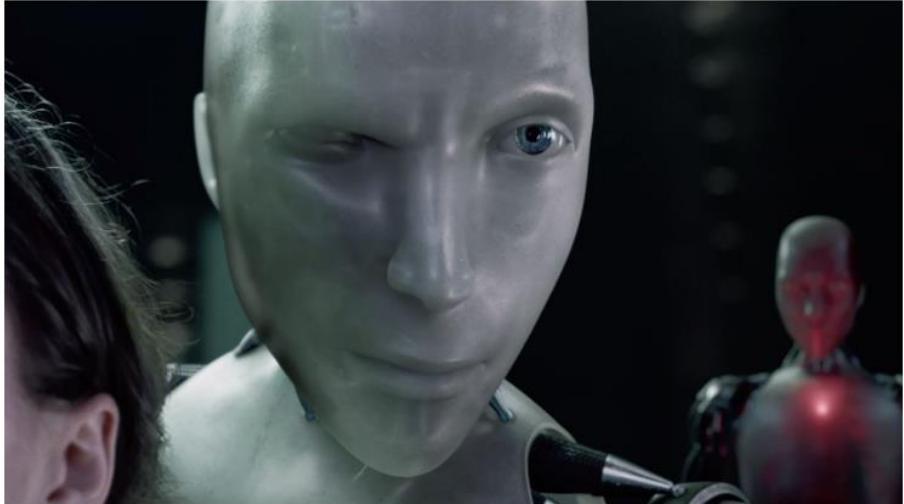
Kimi K2 (1 trillion)





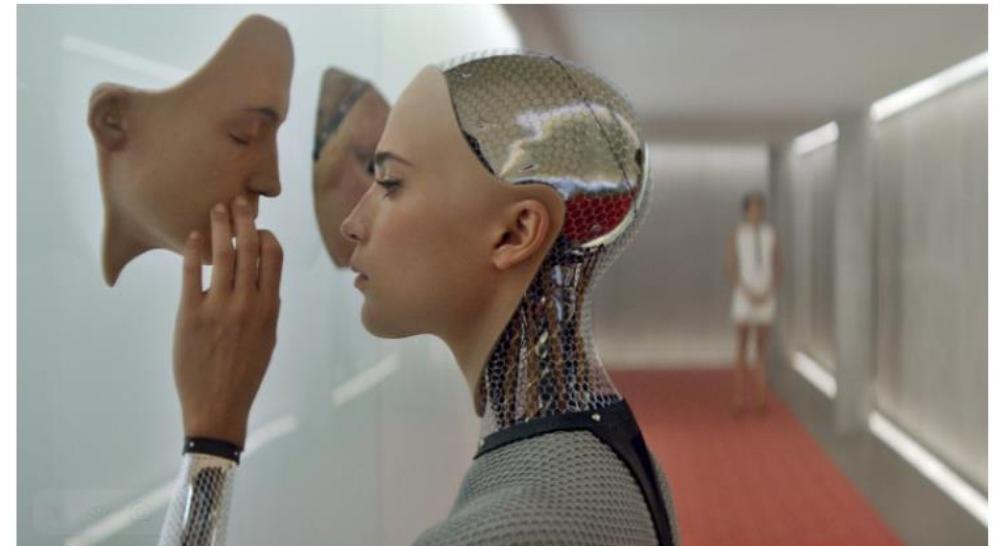
7. Будущие тренды ИИ

Искусственный суперинтеллект (ASI)



Система сможет решать проблемы, которые веками озадачивали человечество, от лечения болезней до раскрытия тайн Вселенной. Она потенциально может произвести революцию во всех областях человеческой деятельности, от науки и техники до искусства и философии. Развитие ASI также представляет собой глубокие экзистенциальные риски, поскольку интеллект, выходящий далеко за пределы человеческого понимания, может быть трудно или невозможно контролировать или привести в соответствие с человеческими ценностями.

Самосознующий ИИ



Развитие действительно самосознующего ИИ поднимет глубокие философские, этические и юридические вопросы. Будут ли такие существа иметь права? Как они будут относиться к людям и другим ИИ?



7. От копирования человеческого разума ИИ переходит к созданию "альтернативного интеллекта"

Человеческое мышление тормозит искусственный интеллект

AlphaGo использовал технику, называемую самостоятельным игровым обучением с подкреплением, чтобы сформировать собственное понимание игры. Метод системных проб и ошибок в миллионах, даже миллиардах виртуальных игр. AlphaGo уверенно победил многократного чемпиона мира, мастера го Ли Седоля в 2016 году, используя странные ходы, которые были бы невероятно редки у человека-противника, и, действительно, это **развило наше, человеческое понимание игры.**

Понятие «интеллект», куда шире, чем может показаться на первый взгляд.

Deepmind начал доминировать в играх, включая сёдзи, Dota 2, Starcraft II и другие, после того как отказался от подражания логики человека. Концепция, что человеческий опыт и интеллект – это лучшее решение для развития, оказалась ошибочной.



7. От копирования человеческого разума ИИ переходит к созданию "альтернативного интеллекта"

AlphaZero не понимает и не видит шахматы так, как Магнус Карлссен. Он никогда не слышал о ферзевом гамбите и не изучал великих гроссмейстеров. Он просто очень много играл в шахматы и выстраивал собственное понимание игры. Основываясь на холодной, жесткой логике побед и поражений, на бесчеловечном и непостижимом языке, который он сам создал в процессе эволюции.

В результате эта модель искусственного интеллекта настолько превосходит любую модель, обученную людьми, что можно с абсолютной уверенностью сказать: ни один человек и ни одна модель, обученная на основе человеческого мышления, не сможет победить в шахматной партии, если ей будет противостоять агент, взращённый на обучении с подкреплением.

Новая модель OpenAI o1 отходит от принципов человеческого мышления



7. Эволюция ИИ входит в фазу, где он создает свою "алгоритмическую природу", отличную от физической

Освобожденные от грубых человеческих размышлений Ньютона, Эйнштейна и Хокинга, перерожденный искусственный интеллект будет использовать подход AlphaGo к пониманию мира. Он будет снова и снова взаимодействовать с реальностью, наблюдая результаты и выстраивая собственные теории на собственных языках о том, что происходит в мире, что невозможно реализовать и почему.

Новые модели искусственного интеллекта не будут подходить к реальности так, как это делаем мы или животные.

«Свободен ото всех оков» в виде нашего языка и мышления, искусственный интеллект выйдет за границы наших знаний и откроет истины существования Вселенной или предложит новые концепции в технологиях, с которыми мы бы не столкнулись изза миллиард лет.

Из хороших новостей – у нас есть некая отсрочка. Это не произойдет в течение нескольких дней или недель.



Эволюция архитектур нейронных сетей от перцептрана к GPT

Руденко Марина Анатольевна
rudenko.ma@cfuv.ru

Кандидат технических наук, доцент кафедры компьютерной инженерии и моделирования
Директор Центр искусственного интеллекта и анализа больших данных ФГАОУ ВО
«Крымский федеральный университет имени Вернадского» Симферополе, Россия.